1991年4月1日発行（毎月1回1日発行）第10巻4号通巻108号　昭和58年11月2日第三種郵便物認可

特集 人とゲームのインタフェイス

決定！1990年度 GAME OF THE YEAR

アマチュアCGAコンテスト入選作品発表

新連載 SX-WINDOW解読術／吾輩はX68000である

4

1991

SOFT BANK オー／エックス

定価560円

SHARP

X68000 SUPER登場

このたび新たにラインアップされた“X68000 SUPER”は、すでに発売されている“SUPER HD”と同様、SCSIインターフェイスを標準装備しています。また、その他のシリーズにはオプションとしてSCSIボード(CZ-6BS1)がサポートされ、大容量外部記憶装置をはじめ、各種SCSI装置との接続が可能になったのは、ご存じのとおりです。

SCSI規格とは……

SCSIは1986年にANSI(米国規格協会)で規格化された仕様で、Small Computer System Interfaceの略。小型コンピュータの周辺機器接続のための世界共通の規格です。大容量外部記憶装置(大容量ハードディスク、CD-ROM、DATなど)に加え、登場が期待される高速スキャナ、次世代プリンタなどのSCSI装置を、デイジーチェーン方式で最大7台まで接続可能。大容量データの高速転送、および単一のインターフェイスでの周辺機器の複数制御が特長です。

X68000と大容量メディア

サウンドクリエーション&コンピュータグラフィックス。X68000のオハコともいうべきこの領域は、感性あふれるユーザーにとって最も魅力的である反面、表現の繊細さに比例して必要な外部記憶容量も増大します。サンプリング、MIDI、レイトレ……。その潜在能力をフルに引き出すには、大容量メディアへの対応が必須です。たとえば、新発売の光磁気ディスク(CZ-6MO1)と光磁気ディスクカートリッジ(JY-701MPA)なら、ディスク1枚で65,536色画像にして1,000枚強、15.6kHzの音声サンプリングデータで約20時間強もの情報を記憶できます。絵に書いた餅とされていた「画像データベー[ス]も、「AD PCMデータライブラリ」さえも、実用レベル。SCSIの採用が、夢の大容[量]メディアに応えてくれるからです。

▲MOディスクドライブ情報

▲MOドライブディレクトリ情報

▲MOディスクのフォーマット

●デイジーチェーン方式による接続例

SCSIボードCZ-6BS1▶
標準価格29,800円(税別)

X68000の先見性

初代X68000は、すでにハードディスクイ[ン]ターフェイスを内蔵していたこと。当時ま[だ]一般的ではなかったハードディスクに対[し]て先見の発想で臨んでいたわけです。今[回]のSCSI対応も同様、100MBを超える大[容]量メディアハンドリングがスタンダードにな[る日]も、そう遠くはありません。

大容量ハードディスクか? 光磁気ディスクか? それとも……

考えもしなかった新しいデバイスか。新製[品]X68000SUPERのSCSIインターフェイ[スに]何を接続するかは、賢明なユーザー[諸]兄にお任せするとして…。このマシンがまた[新]たな一歩を踏み出したことに異論はない[は]ずです。蛇足ながらこのSUPERシリーズ[に]関していわせてもらえれば、その日から大容[量]ハンドリングをお望みの方にはSUPER H[D]を。未来に夢を託したユーザーはSUPE[R]といったところでしょうか。

＊SCSI装置をご使用の場合は、Human68k Ver2.0以上でご[使用]ください。
＊ビジュアルシェル上からはSCSI装置はご使用になれません。

特集　人とゲームのインタフェイス

1990年度　GAME OF THE YEAR

第３回アマチュアCGAコンテスト

SLANG用カードゲームDOBON

(で)のショートプロぱーてぃ

メルヘンメイズ

Oh!X

CONT

〈スタッフ〉
●編集長／前田　徹　●副編集長／植木章夫　●編集／岡崎栄子　浅井研二　●協力／有田隆也　中森　章　後藤貴行　林　一樹　荻窪　圭　華門真人　毛内俊行　吉田賢司　影山裕昭　相馬英智　古村　聡　村田敏幸　丹　明彦　三沢和彦　長沢淳博　宮島　靖　金子俊一　浦川博之　山田純二　●カメラ／杉山和美　●イラスト／永沢しげる　山田晴久　小栗由香　●アートディレクター／島村勝頼　●レイアウト／元木昌子　AD GREEN　●校正／グループごじら

表紙絵：須藤　牧人

1991 APR. 4

E N T S

■広告目次

お望みのパワーシステムへ。

シャープペリフェラルファミリー
X68000

ボード

拡張メモリ

B増設RAMボード
Z-600C専用)
Z-6BE1
準価格35,000円(税別)

B増設RAMボード
Z-601C/611C/652C/
C/662C/663C用)
Z-6BE1B
準価格28,000円(税別)

B増設RAMボード※6
Z-6BE2
準価格79,800円(税別)

B増設RAMボード※6
Z-6BE4
準価格138,000円(税別)

インターフェイス

SCSIボード※7
CZ-6BS1
標準価格29,800円(税別)
(ソフトウェア(SCSIユーティリティ)同梱)

ユニバーサルI/Oボード
CZ-6BU1
標準価格39,800円(税別)

GP-IBボード
CZ-6BG1
標準価格59,800円(税別)

増設用RS-232Cボード
(2チャンネル)
CZ-6BF1
標準価格49,800円(税別)

数値演算プロセッサ

数値演算プロセッサボード
CZ-6BP1
標準価格79,800円(税別)

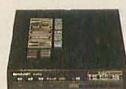
FAX

FAXボード
CZ-6BC1
標準価格79,800円(税別)

MIDI

MIDIボード
CZ-6BM1
標準価格26,800円(税別)

ネットワーク

モデム

モデムユニット※8
CZ-8TM2
標準価格49,800円(税別)
(RS-232Cケーブル同梱)

RS-232Cケーブル

RS-232Cケーブル
(平行接続型)
CZ-8LM1
標準価格7,200円(税別)

RS-232Cケーブル
(クロス接続型)
CZ-8LM2
標準価格7,200円(税別)

LANボード

LANボード
CZ-6BL1
標準価格268,000円(税別)
(イーサネット用)

CZ-6BL2
標準価格298,000円(税別)
(イーサネット/チーパネット両用)
※電源ユニット・ソフトウェア
(ネットワークドライバVer1.0)同梱

入力

インテリジェントコントローラ
CZ-8NJ2
標準価格23,800円(税別)

マウス・トラックボール
CZ-8NM3
標準価格9,800円(税別)

トラックボール
CZ-8NT1
標準価格13,800円(税別)

マウス
CZ-8NM2A
標準価格6,800円(税別)

ジョイカード
CZ-8NJ1
標準価格1,700円(税別)

その他

拡張スロット

拡張I/Oボックス(4スロット)
(CZ-600C/601C/602C/603C/604C/611C/612C/613C/623C用)
CZ-6EB1-BK
CZ-6EB1
標準価格88,000円(税別)

スピーカー

アンプ内蔵
スピーカーシステム(2本1組)
AN-S100
標準価格36,600円(税別)

システムラック

システムラック
(CZ-600C/601C/602C/603C/604C/611C/612C/613C/623C用)
CZ-6SD1
標準価格44,800円(税別)

★印の商品は在庫僅少です。

5,000円(税別・CZ-600C用)、CZ-6BE1B 標準価格28,000円(税別・CZ-601C、CZ-611C、652C、653C、662C、663C用)を増設してください。 ※7 CZ-600C、601C、602C、603C、611C、612C、613Cに装着の場合、I/Oスロット2に装着ください。
Z-652C、653C、662C、663Cに装着の場合はI/Oスロット4に装着ください。また、CZ-6BG1、6BU1、6BL1、6BL2、6BN1などのボードは、接続コネクタとの関係で本ボードとの併用はできませんのでご注意ください。なお、本ボードはX68000用OS Human
3K ver.2.0以上にてご使用ください。 ※8 モデムユニットCZ-8TM2に同梱のソフトはX1/X1ターボシリーズ用です。

器事業本部液晶映像システム事業部第2商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表)

SHARP

ハイアビリティを実証する多彩なソフトウェア。

パソコン通信も、エディタも――。【メモリ常駐型】の優れもの。

Telepotion PRO-68K

CZ-258BS 3月発売予定

他のソフトウェアを実行中でも任意に呼び出して使える メモリ常駐型 のソフトウェアです。パソコン通信/エディタ/カレンダー/スケジュール/住所録/メモ帳/関数電卓の機能を文具感覚でお使いいただけます。「シャープ電子手帳」のデータを「X68000」で簡単に入力・編集することができます。

パソコン通信

●通信速度300から19200BPS/ハードフロー制御(CTS,RTS)対応/高速逆スクロール機能/行入力機能●オートログインなどに対応した本格通信マクロ言語機能/自動オートログインプログラム作成機能●Xmodem, Ymodem, TransIt 2プロトコル対応/漢字コード(シフトJIS,PC漢字,新JIS,旧JIS,DEC漢字)●ANSIターミナルモード,VT-100ターミナルモードのサブセット対応

エディタ

●逆スクロール画面(バックログ)の文章を切りとり編集可能●マルチファイル編集対応/通信時に同時編集可能

カレンダー

●予定の有無を確認できる予定マーク機能/休みの日などの特定日を反転表示する機能●予定日を指定し、その日のスケジュールを表示可能

スケジュール

●他のソフトウェア実行中でも予定の日時を知らせるアラーム機能●必要な日付/時刻/内容を探し出す文字列検索機能●入力データは日付順に自動整列

住所録

●複数の住所録をまとめたときに便利な重複データ消去機能●必要な氏名/索引/電話番号/住所を探し出す文字列検索機能●入力された電話番号データによるオートダイヤル機能/項目ごとの整列機能

メモ帳

●先頭/最終ページへのジャンプ、行挿入/削除/復活、文字列の検索/置換機能

シャープ電子手帳とデータ交換

●専用アダプタを付属(別売の電子手帳通信用ケーブルCE-300Lが必要です。)●前面のジョイスティックコネクタより、簡単な操作でデータ交換可能●電子手帳のレイアウトを確認可能

■通信/エディタ/スケジュール/住所録/メモ帳のデータは、OS(Human68k)対応プリンタで一覧印字が可能です。■対応モデム CCITT V.25bisモデム(CZ-8TM1、CZ-8TM2など)ATモデム(MD24FS7、AX/2400、PV-A2400、SR-120Sなど)■他のソフトウェア中で本ソフトを起動する場合、本体メモリの空き領域として最低270Kバイトが必要です。■同時実行可能なソフト例(◎は本体メモリ2Mバイト必要です。)●Communication PRO-68K◎CARD PRO-68K◎DATA PRO-68K◎BUSINESS PRO-68K◎Ccompiler PRO-68K◎TOP給与計算エキスパート◎TOP財務会計●THE福袋V2.0◎NEW Print Shop PRO-68K◎本体同梱ワープロ●X-BASIC●エディタ(ED.X)◎AI-68K(Staff LISP/OPS PRO-68K)●FAXツール◎CYBERNOTE PRO-68K◎Hyperword◎CANVAS PRO-68K●スキャナツール■対応シャープ電子手帳は、PA-8800/8500/7500/7000/6500/6000です。

※Stationary PRO-68k(CZ-240BS)を既にお持ちの方には、有償バージョンアップサービスを行います。

さらにパワーアップしたMIDIマルチトラックレコーディングソフト

Musicstudio PRO-68K

CZ-261MS 標準価格28,800円(税別)

MIDI、内蔵FM音源、内蔵ADPCM音源の各出力を同期演奏・編集が可能。ますます拡がるミュージックワールド。
MUSIC PRO-68K(MIDI)との双方向データコンバート機能。演奏した曲が楽譜に。
トラック単位、指定範囲(パターン)のデータセーブ機能搭載で、さらに使いやすくなりました。

スタジオ感覚のMTRウィンドウ(24トラック)

豊富なリアルタイムレコーディング機能

●16チャンネルマルチレコーディング、パンチイン・パンチアウトもサポート。●4種類の分解能(最高♩=240、編集中の変更可能)に加えて、クオンタイズ機能もサポート。

強力なステップ入力・ステップエディット機能搭載

●マウス・X68000キーボード・MIDIキーボードによるスピーディな入力。●データのプリントアウトもサポート。●15行表示のステップウィンドウ、再生時のデータシンクロスクロール。●スピーディなタイ入力、連符入力は15連符までサポート。

充実のリアルタイムミキシングエディット機能搭載

●ボリュームデータ、パンポットデータのリアルタイム録音。●フェードイン・アウトがコントロールできる2種類のグループフェーダーや、オートフェーダー・オードパンポットエフェクト機能搭載。●各トラック毎に再生用ベロシティフェーダーと、再生用エフェクトキャンセルスイッチをそれぞれ搭載。

ソングポジションポインタ送受信対応

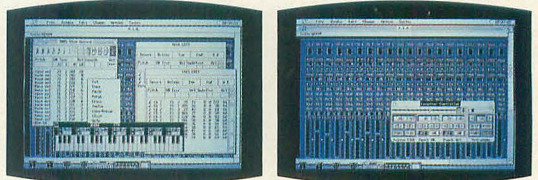
●外部機器との同期演奏をマスター、スレーブ時ともサポート。

演奏しながらウィンドウ内でテレビや絵を呼び出せるビジュアルイン・ウィンドウ機能

使用可能画像フォーマット:GM0, GL0, GS3, GM3, GL3, EMX, EM3, PL3, PLM PIC

※テレビ画像を楽しむには、カラーイメージユニットCZ-6VT1が必要です。

国本佳宏氏のオリジナルサンプル曲及び長谷川一光氏の画像データを収録

※Musicstudio PRO-68K(CZ-237MS)及びMusicstudio PRO-68K ver.1.1(CZ-252MS)を既にお持ちの方には、有償バージョンアップサービスを行います。
※本ソフトの動作にあたっては、MIDIボード(CZ-6BM1など)が必要です。

シューティングゲーム

中華大仙

CZ-268AS
標準価格7,900円(税別)

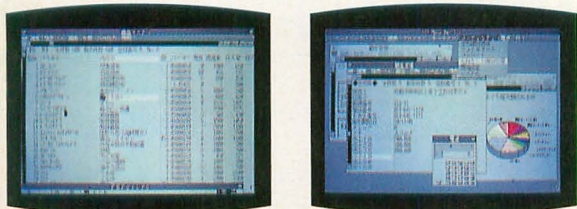
シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部液晶映像システム事業部第2商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表)へ

望みのワークベンチへ。

高速・新マルチウィンドウ搭載。一覧表入力や多彩なグラフ機能も装備。

CARD PRO-68K ver2.0

CZ-253BS 3月発売予定

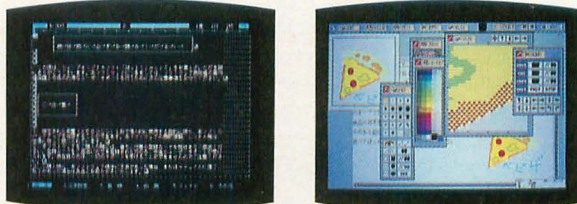
①新マルチウィンドウシステム搭載により操作性の向上、スピードアップ。キーボード操作に対応。②従来から要望の高かった一覧表画面での入力をサポート。③グラフ機能搭載。④キーマクロ機能搭載。⑤斜め罫線に対応するなどワープロ機能の表現強化。⑥プログラム機能の強化。⑦子プロセスにより、Human68kのコマンドを実行可能。

〈大容量、高速データ処理〉●最大処理件数100万件(レコード)、1枚のカード項目、最大999項目、1項目長255バイト、1カード最大文字数32767文字の大容量データ処理が可能**〈検索機能〉**●B-TREE ISAMとバッファリング検索を使用した高速マルチ検索●以上、以下、超過、未満、完全一致、不一致、先頭一致、先頭不一致、包含、不包含の条件の他、AND、ORの論理演算子も簡単に複数設定可能。●ワイルドカード検索●マルチ検索条件をディスクに登録可能**〈整列機能〉**●JISコード/五十音の昇順/降順ソート●複数の項目に対する高速マルチソート●マルチソート条件をディスクに登録可能**〈グラフ機能〉**●折れ線、2次元分布、縦棒、円、レーダーチャート、折れ線・マーク、折れ線・縦棒の7種類をサポート●カラー/白黒の表示切り替え可能**〈ワープロ機能〉**●ワープロ機能で作成したフリーレイアウト画面でデータ入力、検索表示、印字が可能●罫線は8種類(実線,破線,一点鎖線,点線のそれぞれ細線,太線)の矩形/斜め線罫線をサポート●文字飾りは、右寄せ、左寄せ、センタリング、強調、上つき/下つき1/4倍角、横倍角をサポート●他のレイアウト画面/テキストファイルを参照可能●範囲先指定で、CUT&PASTEも使用可能**〈プログラム機能〉**●豊富な関数群(58種)、命令(62種)を用いて複雑な処理が可能。計算式は登録時にコンパイルされ高速実行●項目間演算、串刺しデータ間計算をサポート●自動計算式として登録することによりデータ入力時にリアルタイムな計算が可能**〈電卓機能〉**●どのウィンドウ使用中でも呼び出し可能●計算値をウィンドウへ貼り付け可能**〈印刷機能〉**●文書印刷だけでなく一般帳票や葉書への差込み印刷、システム手帳のリフィル印刷をサポート**〈キーマクロ機能〉**●キー操作を記録し複雑な操作も簡単に実行可能**〈コンバート機能〉**●他の多くのソフトと双方向のデータ互換を実現「BUSINESS PRO-68K」「Multiplan」「Lotus 1-2-3」などの表集計ソフト、「一太郎」「Hyperword」などのワープロソフト、「Ninja」「隼 HAYABUSA」「1-2-3Card」「CARD PRO-68K ver.1.0」などのデータベースソフトに対応(区切りあり・区切りなしASCIIファイル、CSVファイル、SYLKファイル、「1-2-3Card」「隼」ファイル)

*「Kamikaze」「Ninja」は㈱サムシンググッドの、「Multiplan」は米国マイクロソフト社の、「Lotus 1-2-3」はロータス㈱の、「一太郎」は㈱ジャストシステムの、「隼 HAYABUSA」「1-2-3Card」はダットジャパン㈱の登録商標です。*CARD PRO-68K(CZ-226BS)をすでにお持ちの方は有償バージョンアップを行います。
* 本ソフトの動作には、メインメモリ2MBが必要です。

ウィンドウでWYSIWYG編集。カラーグラフィック、高速テキストモードもサポート。

マルチワープロ PRO-68K Multiword

CZ-225BS 3月発売予定

①WYSIWYGな編集が行えるウィンドウモードと素早い編集が行えるテキストモードをサポート。②グラフィックを文章中に自由にレイアウト可能。③同一文章中で複数の改行幅指定など多彩な機能をサポート。④高品位のプリントアウトができるレーザープリンタ、カラー印字(8色)をサポート。

〈ウィンドウモード〉●マルチウィンドウによる複数文章(最大10個)の同一画面上での編集●レイアウト表示機能/外字作成機能(16, 24)/辞書ユーティリティ●ウィンドウのアイコン化機能/ショートカット機能●電卓機能(文章中に計算結果を全角/半角で入力可能)●キーリピート速度/ブリンク速度調整可能**〈テキストモード〉**●エディタ感覚で素早い編集が可能●スムーズなウィンドウモード/テキストモード切り替え**〈グラフィック機能〉**●任意の場所に何回でもレイアウト可能●拡大/縮小、ルーペ、ペン先編集、ボックス/円、ペイント機能●カット/コピー/移動、反転/回転/対象コピー、スケール表示、タイル編集●イメージスキャナをサポート**〈書式指定〉**●同一文章中での複数の改行幅指定(1/2改行、1/3改行、1/4改行、0改行、ミリ単位の自由改行)●自由な用紙サイズ(A4、B4、B5、10インチ、15インチ、葉書、ミリ単位の自由設定)**〈拡大文字表示機能〉**●最大16倍角までの任意の文字拡大、スムージング機能**〈文字種、文字飾り機能〉**●半角、全角、上つき/下つき1/4角文字、文字色指定、回転、アンダーライン、かすれ、網掛け、強調、リバース、特殊文字**〈罫線機能〉**●罫線固定/移動モード、保護モード(左右)をサポート●外線、内線、ボックス、十字、移動、消去、罫線種変更**〈文字列変更機能〉**●削除/保存/貼付/復活/検索/置き換え**〈行揃え機能〉**●改行幅変更、文章幅変更、右寄せ、左寄せ、センタリング**〈カーソル移動〉**●文頭/文末、行頭/行末、任意の位置、マーク、他の文章への移動**〈印刷機能〉**●ページ指定印刷、部分印刷、一括印刷、縦書き印刷、袋閉じ印刷、段組印刷、ヘッダー印刷、フッター印刷**〈他のソフトウェアとのデータ互換〉**●本体同梱ワープロ/「一太郎」の文章ファイル読み込み●テキストファイル読み込み/書き出し

*「一太郎」は㈱ジャストシステムの、登録高標です。
* 本ソフトの動作には、メインメモリ2MBが必要です。

Multiwordその他新作ソフト講習会開催
※お問い合わせは各ショールームまで。

会場	日程	電話
●東京OAショールーム	3月30日(土)・31日(日)	TEL(03)3260-8365
●横浜OAショールーム	3月28日(木)	TEL(045)201-6525
●名古屋OAショールーム	3月26日(火)	TEL(052)323-5147
●大阪OAショールーム	3月23日(土)	TEL(06) 222-7655
●神戸OAショールーム	3月22日(金)	TEL(078)291-8715
●福岡OAショールーム	3月23日(土)	TEL(092)481-2860

〈受講料〉2,000円(税別)

アッパレ
X68000EXEクラブ
EXEおみこし活動に参加しよう!
◉まずはEXEクラブへ◉
入会無料で3つのメリット!手続きは本体同梱の入会申込ハガキを送るだけ。
メリット1▶会員番号入りオリジナル会員証電卓がもらえます。
メリット2▶各種フェアご優待・イベント案内等、数々の特典があります。
メリット3▶X68000の活用情報が手に入る「EXEおみこし活動」に参加できます。
※「入会申込ハガキをなくしてしまった」という方は、おみこし活動隊までお電話下さい。
◉EXEおみこし活動とは?◉
いわば「X68000ユーザーの、X68000ユーザによる、X68000ユーザーのための」活動です。おみこしPRESSを通じて会員同士情報を交換し、もっと68を使いこなして盛り上がってしまおう!というワケ(モデムがなくてもできるパソコン通信のようなもの?!)なので、X68000へのラブコール、会員独自のテクニック、活用法(マニアックなものでなくても他の会員には貴重!)等あなたの68自慢をドシドシ聞かせてください。会員からのメッセージは「おみこし活動隊」が整理してコミュニケーションペーパー「おみこしPRESS」にバッチリ掲載します。
おみこし活動隊へのアクセス方法
投稿受付/大阪市淀川区西中島1丁目9-16 新大阪ストロングビル2F X68000EXEクラブ「おみこし活動隊」係
電話受付 06-886-0354 FAX受付 06-304-1539
★受付時間……平日/昼2時〜夜8時 日祝/昼12時〜夜8時
◉そしてEXE会員究極の◉
「おみこしかつぎ人」を大募集!
「かつぎ人」とは、より積極的におみこし活動に参加する人のこと。EXE会員は「かつぎ人」になることで、X68000ユーザーとしてますます充実、3つのメリットで強力にサポートされます。
《メリット1》「おみこしかつぎ人の集い」に参加できます。シャープとEXE会員の双方向コミュニケーションの場として開設されるX68000情報交換会「おみこしかつぎ人の集い」は、シャープの68スタッフと直に意見交換ができるおいしいチャンス。68の最新情報満載のこの会に参加すれば、68ユーザーとしてトップレベルです。
《メリット2》「おみこしPRESS」定期送付。お店まで足を運ばなくても「おみこしPRESS」が毎号お手元に届きます。
《メリット3》「ソフトウェア・フィールド」直送。X68000最新ソフト・各種周辺機器が一覧できる「ソフトウェア・フィールド」を半年に1回お送りします。
おみこしかつぎ人になるには……
以下の年会費(おみこしかつぎ代)が必要です。〈個人入会〉3,000円/〈グループ入会(5人1組)〉2,500円/1人(郵便振込にて申込受付)
★詳細は店頭の「おみこしPRESS」をご覧になるか、または「おみこし活動隊」にお電話ください。
Multiwordその他
新作ソフト講習会開催
X68000をサポートする多彩なソフトウェア。その性能・魅力を知っていただこうと、各地ショールームにて講習会を開催いたします。新作ソフトを体験していただくため、会場ではソフトを実際に操作していただきます。最寄りのショールームへぜひおいでください。
〈使用ソフト〉
●マルチウィンドウ編集・グラフィックス機能搭載等、話題の本格ワープロ
「マルチワープロ PRO-68K Multiword」
●高速新マルチウィンドウ搭載カード型データベース
「CARD PRO-68K ver.2.0」
●期待のSX-WINDOW対応グラフィックツール
「Easypaint SX-68K」
※受講料として2,000円が必要です。詳細は各ショールームまでお問い合わせください。
東京OAショールーム 3月30日(土)・31日(日) ☎(03)3260-8365
シャープ東京支社ビル
横浜OAショールーム 3月28日(木) ☎(045)201-6525
横浜三菱商事ビル内
名古屋OAショールーム
シャープ名古屋ビル
大阪OAショールーム 3月23日(土) ☎(06)222-7655
日本生命今橋ビル内
神戸OAショールーム 3月22日(金) ☎(078)291-8715
日本生命三宮駅前ビル内
福岡OAショールーム
博多駅前第一生命ビル内
ヨシ!
シャープ株式会社 ●お問い合わせは…シャープ(株)電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22

C-TRACE シリーズ最強

C-TRACE+（プラス）

価格￥198,000(税別)

対応機種　PC-9801シリーズ
PC-286、386シリーズ
X68000シリーズ

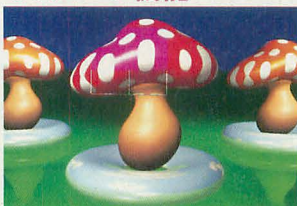
●メタボール
機的な質感を表現 他のプリミティブとの論理演算が可能

●ワイヤービュー搭載

てのプリミティブをワイヤーフレーで表示
ンダリング前の形状確認OK

●αチャンネル機能

より高度な画像合成が可能
α情報をスーパータブロープレミアム(サピエンス社)で使用可能

●スコープ機能

部分的に画像の再計算が可能
アニメーションへの応用で作業を大幅に効率化

※写真は広告用に撮影したものです。

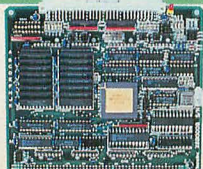
- ●スポット光源への対応
  照射範囲を自由に設定。境界のぼかしも可能
- ●ボクセル分割による高速レンダリング
- ●ツリー構造を採用したデータ記述
- ●アニメーションへの対応
- ●豊富なマッピング
- ●豊富なプリミティブ
- ●プリミティブ同士の論理演算
- ●移動、回転、縮小拡大、自由自在

超高速！メタボール対応

NEW

C-TRACE TP＋（プラス）★

価格￥398,000(税込)

●PC-9801シリーズ、PC-286、386シリーズ、X68000シリーズ ●高速なレンダリング処理をメタボールにも実現(80386CPU約40倍、68000CPU約170倍) ●トランスピュータボードとC-TRACE＋をセットアップ●TP Ver.3.0と差額交換受付中

超高速

C-TRACE TP Ver.3.0★

価格￥298,000(税込)

●PC-9801シリーズ、PC-286、386シリーズ、X68000シリーズ ●ワークステーション並みのスピードをパソコンでも実現したレイトレーシング ●並列処理でさらにスピードアップ ●トランスピュータボードとC-TRACE Ver.3.0のセット

メモリー解放宣言

C-TRACE98 EXTENDER

価格￥128,000(税別)

●PC-9801シリーズ、PC-286、386シリーズ●メインメモリとして最大16M使用可能●EMSによるメモリ拡張のようにスピードを犠牲にしません●30%の高速化(当社Ver.3.0比)

長谷川一光 CG塾 4月開講
お問い合せ先 〒101 千代田区神田岩本町4番地 コアスタジオ TEL.03-3256-7035

★の製品は店頭販売しておりません。直接当社までお申込み下さい。

●株式会社キャスト ●お問い合せ先 〒158 東京都世田谷区等々力2-1-13 TEL:03-3705-1065 FAX:03-3705-5224

シューティングゲーム攻略のために生まれたスーパーアイテム“メンクリ”

MENKURI

バトルシート“メンクリ”の特徴

- 市販のジョイスティックをガッチリ固定して操作性・操作フィーリングを向上させます。
- テレビ画面からある程度の距離が保てるため、視力の低下を防ぐことができます。
- パソコンでゲームをするさい、キーボードの避難場所に困りません。
- 実用新案登録出願済

お問い合わせ電話番号 **0256(33)6111**—アイアンクラフト

●写真のジョイスティックは商品には含まれません。
●改良のため予告なく仕様を変更することがあります。

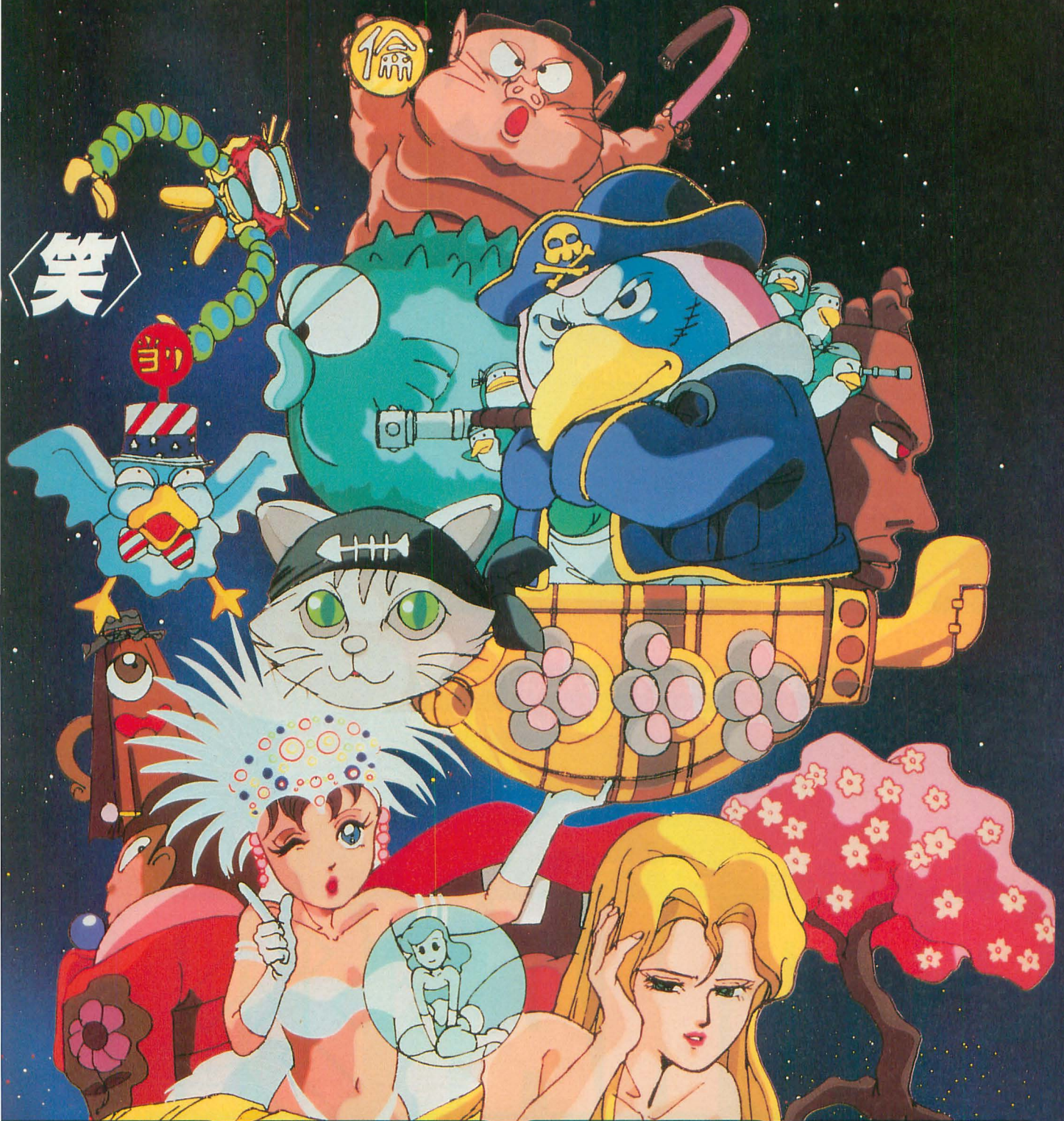
定価7,800円（税込）送料1,000円（日本国内一律）

＊通信販売でのみ取り扱っておりますので、とりあえず電話でお問い合わせ下さい。
　受付時間 AM10：00～PM6：00　毎週火曜日定休

＊FAXでのお申込は24時間OK。 FAX 0256(33)6110

＊代金のお支払い方法。
1. 下記の住所へ現金書留でおくる。
2. 郵便振替を利用する。
（新潟6-23194 アイアンクラフト）
3. 銀行振込を利用する。
（大光銀行 東三条支店　普通口座 463714 アイアンクラフト）
4. 代金引き替えで商品を受け取る。

〒955 新潟県三条市田島2丁目23番3号　田島ハイツ203号室
アイアンクラフト

HOME DATA
死ぬまで転がせ!
不眠不球の面白さ
MARBLE MADNESS
マーブルマッドネス
対応機種
68000 シリーズ 発売中 (5″2HD)
C-98 VM以降 3月30日発売(5″2HD 3.5″2HD)
は要16色ボード・要640KB)
TOWNS シリーズ 4月下旬
各 ¥9,700(消費税別)
© ATARI GAMES CORP © TENGEN INC © TENGEN LTD
〈プログラマー募集〉
センブラ、Cでプログラムの組める人、経験者優遇
細はTELにて。 担当 倪(ニイ)まで
HOME DATA
株式会社ホームデータ
〒651 神戸市中央区葺合町馬止1-10 ホームデータビル
Phone:078(261)2790(代) FAX:078(261)2792

未来とは定められ

人類の歴史は偶然の結果の記録ではない。
それは、定められたひとつの目的にしたがって操作された結果である。
あらゆる予言の書の存在…。
なぜ人が未来を知り得るのであろうか?
人類は定められた運命を変える事ができるのか?

X68000 ONLY 5'2HD(5枚組) **価格¥12,000(税抜)**

■マウスオペレーションで簡単操作　■200枚を超える美しいグラフィック
■史実の謎に迫る野心的ストーリー　■現地取材をもとにしたリアルな構成

シグナトリー

スペキュレーティブ・アドベンチャー

SIGNATORY

―調印者―

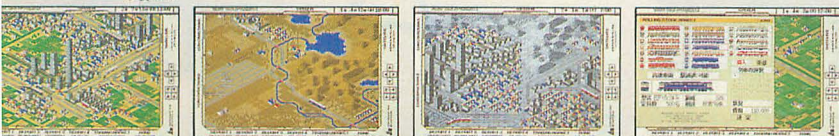
提供■NCS　制作総指揮・総監督・原作■鈴木 力　脚本■成田伸子　出演■ケニー・フィリップ/バーバラ・ドゥーディ/トーマス・スウェイジ他　制作■Ten

■制作スタッフ■スクリプト/皆川正三 ■SE・プログラム/橋谷利幸 ■メイン・プログラム/Hかずき ■チーフデザイン/石井秀明 ■デザイン/本間繁二郎/大村政幸/矢田 智/古澤雅子 ■音楽・効果音/高橋大昌 ■NY・南米取材/青空風太郎 ■NY取材協力/氷

オリジナルテレカ・プレゼント! 「シクナトリー」をお買い上げ頂き商品内のユーザー・ハガキをお送り下さった方の中から、先着500名様にオリジナルテレカをプレゼント。

た運命なのか？

NCS 日本コンピュータシステム株式会社
〒106 東京都港区西麻布4-16-13 第28森ビル TEL.03-3486-6314（代表）
お問い合わせはソフトウェア・プロダクト部（直通）03-3486-6588（受付時間）9時〜18時

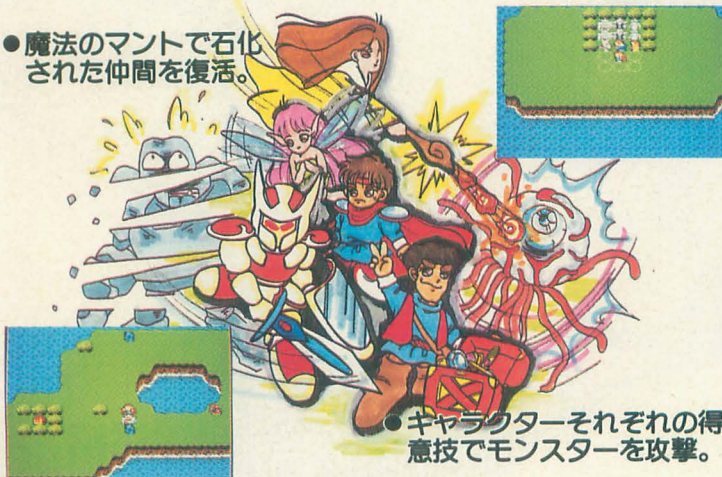
あの「AIII」が、マップコンストラクションとともに、登場。
待望。
ARTDINK
A.III.
SIMULATION GAME
RAILROAD MANAGEMENT
URBAN DEVELOPMENT
A列車で行こう
あなたは鉄道会社のオーナー。まず経営を軌道にのせる。
●基本は安定収益の路線をつくること。効率的な列車運行ダイヤを組むことだ。
●用意された車両は、新型通勤列車「205系」、高速貨物列車「EF66」、最新鋭特急列車「AR-III」など19種におよぶ。特長をよくつかめ。
みるみる広がるビル夜景。巨大都市開発計画に着手する。
●建物の資材をつくる、工場。経済を活性化させる、ビル、デパート、レジャー施設。いずれも都市発展の強力な原動力となるが、無造作な建築は禁物だ。
資金を調達する。エグセクティブの本領が問われる場面だ。
●設備資金の調達に、銀行から融資を受けることも必要だ。
●相場のタイミングを見て、24銘柄の上場株を売買するのもおもしろい。
マップコンストラクション付き。マップデータは他機種間で完全互換。
A列車で行こうIII
X68000
●要2MB
タケルにて発売予定
TAKERU
SOFT WARE VENDING NETWORK SYSTEM
栄冠は君に
高校野球全国大会
激闘開幕。母校の夢は叶うか。
X68000
●5" 2HD(3枚組)
標準価格 9,500円
〒260 千葉市高洲1-21-1三基ビル3F ☎0472-79-9392 株式会社アートディンク

パズルというと、何だかめんどくさそう…なんていう人がいるけれど、そういう人にかぎってやってみると結構ハマったりするのです。

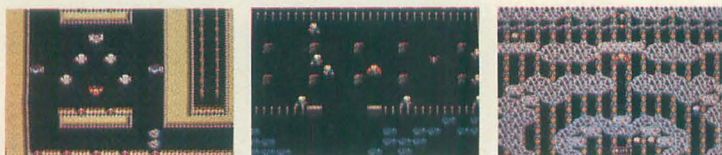
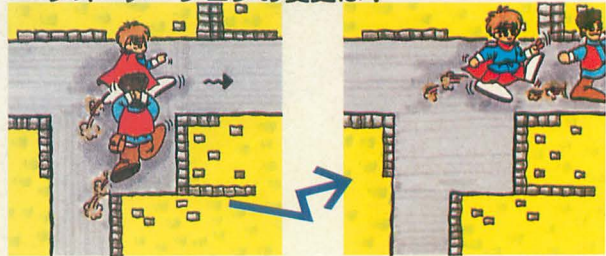
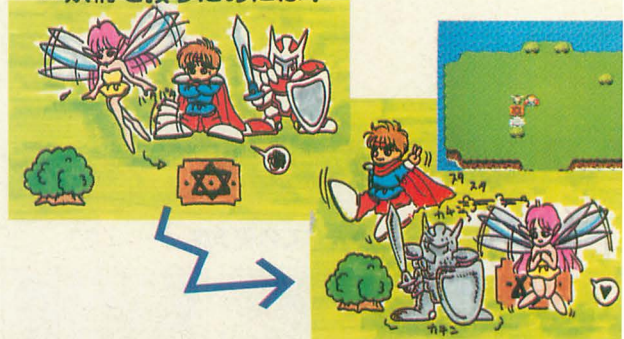
???が💡となったときは、もう ですよね。

4月下旬 発売予定

X68000シリーズ・5"2HD2枚組・¥7,800

邪神の呪いにより、石像に変えられた妖精の魂を神殿に帰すため、アルフレードは戦う！

●魔法のマントで石化された仲間を復活。

●キャラクターそれぞれの得意技でモンスターを攻撃。

●豊富な場面、次々と現れるモンスター。

●フォーメーションの変更は？

●妖精を救うためには？

●通信販売(送料無料)のお知らせ
工画堂スタジオでは通信販売をしております。ご希望の方は、品名・機種名・住所・氏名・電話番号を明記の上、3%の消費税を加算して現金書留でお申し込みください。

●開発スタッフ募集のお知らせ
プログラム(PC-98・PC-88・MSX)・アシスタントプログラム・ゲームデザイン・グラフィックのスタッフを募集中です。御連絡ください。

㈱工画堂スタジオ
〒162 東京都新宿区市谷台町
TEL03-3353-7724

資料請
Oh!x

ゴルフゲームのスタンダード！

LICENSED BY
AUGUSTA NATIONAL GOLF CLUB

NEW 3D GOLF SIMULATION

TM

はるかなるオーガスタ

まもなくプレイオン。

neXt
RPG・ACT・SLG、最強のラインナップで次世代体験……neXt！

観てから遊ぶか、
遊んでから観るか

マスターズ・トーナメントいよいよ開催

4月11日から4月14日まで4日間の熱い戦い。
TBS系でテレビ放映 4月X日 PMX:XX

※写真はX68000版開発中画面です。

オーガスタ・ナショナル
ゴルフ・クラブと正式契約

X68000版 4月5日㈮発売

×68000版特長

- 実際にゴルフコースに立った状態と同じ視野でプレイ可能。
- どの地点にきても全方向の視野画面をリアルタイム3D表示。
- ホールすべてにアンジュレーション(起伏)を3Dで表示。
- ボールの落下地点の状態によってバウンド、転がり等が本物同様に変化。
- ストロボアクションモードでボールの軌跡を確認可能。
- トップスピン・バックスピンも自在。キャディーは4人の中から選択。
- ADPCMによるリアルなサウンド。ショット音、歓声、小鳥のさえずりまでも忠実に再現。
- プレイモードは3種類。ストロークプレイ、マッチプレイ、トーナメントプレイ。
- スコア・各種個人データ等を自動保存、プリントアウトも可能。
- 初心者でも手軽に楽しめるスローモード機能あり。
- 31KHZ/15KHZ両モード対応。

POLYSYS™
Integrated 3D Processor

このマークはT&E SOFTの商標です。
POLYSYSを搭載の3Dソフトには、このマークが表示されます

RPG-neXt………ルーンワース 黒衣の貴公子
ACT-neXt………幻 獣 鬼
SLG-neXt………遥かなるオーガスタ

■PC-9801VMシリーズ(5″2HD 2ドライブ)要640K RAM
■PC-9801UV/UFシリーズ(3.5″2HD 2ドライブ)要640K RAM
■PC-9801N/URシリーズ〈NOTE専用版〉(3.5″2HD 1ドライブ+1RAMドライブ)
※上記のソフトはエプソンPC-286，386シリーズに対応
■FM TOWNS(CD-ROM & 3.5″2HD)要2M RAM

標準価格 各¥12,800(税別)

Technology & Entertainment Software

株式会社 ティーアンドイーソフト
〒465 名古屋市名東区豊が丘1810番地 PHONE:052-773-7770

No Copy
このマークは不法コピー禁止マークです

● 3Dゴルフに関するお問い合わせは、NEW 3D GOLF事務局まで PHONE:052-773-7757

■店頭にて、ゲームソフト25%OFF!!（税別）、超低金利 ハッピークレジットをご利用ください!!
■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい。

厳選された製品を、より安く、より早く、皆様のお手元に!!

広告掲載商品以外の製品も取扱っております。

オクト厳定!! SUPER-HDスペシャルセット～今がチャンス!! 電話で値切ろう（送料無料）

●ザ・ワークステーションと呼ぶにふさわしい

SUPER-HD＝オクト厳定セット!!

※マウス・トラックボール付!!
ディスプレイにはスピーカ2個
チルト台付!!

これでドーダ!! の価格です。マイッたネ。

Ⓐ：CZ-623C-TN＋CZ-606D-TN…………定価合計￥577,800▶大特価
●ドンドン お電話下さい。安心、簡単なクレジットをご利用下さい。

Ⓑ：CZ-623C-TN＋CZ-613D-TN…………定価合計￥633,000▶大特価
●ドンドン お電話下さい。信頼して下さい。この価格です!!

現金特価!! お電話下さい。

上記セットお買い上げの方に、オクトからのプレゼント!!
①MD-2HD 10枚
②シムシティ（ゲームソフト￥8,800）
③ジョイカード（連射式）
④シリコンキーボードカバー（￥2,800）

※超低金利クレジットをご利用下さい。1回～60回払い、頭金ナシ!! ボーナス1回及び2回払いOKです。

X68000ソフト大セール実施中※ゲームソフトオール25%off 送料￥500

〈グラフィック〉●Z's STAFF PRO68K Ver.2.0
（シャフト）定価￥58,000…………特価￥39,400

〈データーベース〉●KAMIKAZE
定価￥68,000…………特価￥45,400

〈グラフィック〉●C-TRACE68Ver3.0
（キャスト）定価￥78,000…………特価￥69,500

〈C言語〉●C & Professional Pack
定価￥58,000…………特価￥41,000

〈グラフィック〉●サイクロ エキスプレス α 68
定価￥97,000…………特価￥73,000

〈グラフィック〉●デジタルクラフト
定価￥39,800…………特価￥28,000

〈ワープロ〉●ハイパーワード
定価￥39,800 CZ-251BS…………特価￥29,800

〈開発ツール〉●C-コンパラPRO68KV.2
定価￥44,800 CZ-245IS…………特価￥33,300

〈CGツール〉●CANVAS PRO68K
定価￥29,800 CZ-249GS…………特価￥22,200

型名	商品	定価	特価
CZ-212BS	BUSINESS PRO-68K	￥68,000	￥48,000
CZ-213MS	MUSIC PRO68K	￥18,800	￥13,500
CZ-214MS	SOUND PRO-68K	￥15,800	￥11,500
CZ-215MS	Sampling PRO-68K	￥17,800	￥12,800
CZ-219SS	OS-9/X68000	￥29,800	￥21,000
CZ-220BS	DATA PRO-68K	￥58,000	￥41,000
CZ-223CS	Communication PRO-68K	￥19,800	￥14,300
CZ-224LS	THE 福袋 V2.0	￥9,900	￥7,500
CZ-226BS	CARD PRO-68K	￥29,800	￥21,300
CZ-241BS	システム手帳リフィル集	￥9,800	￥7,500
CZ-242BS	活用フォーム集	￥9,800	￥7,500
CZ-244SS	Homan 68K Ver2.0	￥9,800	￥7,500
CZ-247MS	MUSIC PRO-68K(MIDI)	￥28,800	￥20,800
CZ-240BS	Stationery PRO-68K	￥14,800	￥11,500
CZ-243BS	CYBER NOTE PRO-68K	￥19,800	￥15,200
EW		￥38,000	￥29,800
G-68K		￥14,800	￥11,400
E-68		￥19,800	￥15,300
CZ-255GS	CANVASドローグラフィックLIB	￥8,800	￥6,600
CZ-256GS	CANVASドローグラフィックVOL.2	￥8,800	￥6,600
CZ-260LS	XBAS to CHECKER PRO68K	￥9,800	￥7,500
CZ-259SS	SX-WINDOW Ver1.0	￥6,800	￥5,000
CZ-234LS	AI-68K	￥188,000	￥139,000
CZ-252MS	MUSIC STUDIO PRO68K	￥28,800	￥21,500

モデムコーナー （送料￥1,000）

オムロン
●MD-1200AIII…………特価￥14,500
●MD-12FS…………特価￥15,000
●MD-24FP4II…………特価￥26,000
●MD-24FN5…………特価￥30,000
●MD-24FS4…………特価￥31,000
●MD-24FS5…………特価￥31,500
●MD-24FS7…………特価￥43,500
●MD-24FC5…………特価￥34,000
●MD-24FP5II…………特価￥28,500
●MD-24FN4…………特価￥27,000
●MD-24FJ4…………特価￥31,000
●MD-24FJ5…………特価￥34,000
●MD-24HS…………特価￥64,000
●MD-48HS…………特価￥98,000
●MD-96FS5…………特価￥131,000

AIWA
●PV-A24VM5…………特価￥30,000
●PV-M24VM5…………特価￥30,000
●PV-A12…………特価￥14,500
●PV-M24…………特価￥28,500

熱転写カラー漢字プリンター （ケーブル用紙付） 送料￥1,000

■CZ-8PC5 NEW

● 48ドット
● 熱転写カラー漢字プリンター
定価￥96,800
大特価TEL下さい!!

①CZ-8PK10（24ピン漢字プリンター136桁）
定価￥97,800……大特価!! TEL下さい。
②CZ-8PG1（24ピンカラー漢字プリンター80桁）
定価￥130,000……大特価!! TEL下さい。
③CZ-8PG2（24ピンカラー漢字プリンター136桁）
定価￥160,000……大特価!! TEL下さい。
④IO-735×（カラーイメージシェット）
定価￥248,000……大特価￥177,000

パソコンラック 推奨 送料無料

①五段キャスター付

5段キャスター付
キーボードが収納できるから、手元でマウス操作がラクラクできる
棚板5段のマルチに活用できるディスク。
ウーン、こいつはデキル!
1325(H)×640(W)×700(D)
特価￥15,000

②四段キャスター付

4段キャスター付
どんなパソコンにもフレキシブルに対応!
使い易いデスクです。
1245(H)×614(W)×600(D)
特価￥11,000

周辺機器コーナー （送料￥500）

型名	品名	定価	特価
●CZ-6BE1	1MB増設RAMボード	（￥35,000）	▶特価￥26,000
●CZ-6BE1B	1MB増設RAMボード	（￥28,000）	▶特価￥21,000
●CZ-6BE2	2MB増設RAMボード	（￥79,800）	▶特価￥60,000
●CZ-6BE4	4MB増設RAMボード	（￥138,000）	▶特価￥103,000
●CZ-6BF1	増設用RS-232Cボード	（￥49,800）	▶特価￥38,000
●CZ-6BG1	GP-IBボード	（￥59,800）	▶特価￥45,000
●CZ-6BM1	MIDIボード	（￥26,800）	▶特価￥20,200
●CZ-6BN1	スキャナ用パラレルボード	（￥29,800）	▶特価￥22,500
●CZ-6BP1	数値演算プロセッサボード	（￥79,800）	▶特価￥60,000
●CZ-6BO1	ユニバーサルI/Oボード	（￥39,800）	▶特価￥30,500
●CZ-6EB1/BK	拡張I/Oボックス	（￥88,000）	▶特価￥65,800
●CZ-6VT1/BK	カラーイメージ・ユニット	（￥69,800）	▶特価￥52,300
●CZ-8NM2A	マウス	（￥68,800）	▶特価￥5,300
●CZ-8NT1	マウストラックボール	（￥98,800）	▶特価￥7,500
●CZ-8NS1	カラーイメージスキャナ	（￥188,000）	▶特価￥137,000
●CZ-6BC1	FAXボード	（￥79,800）	▶特価￥60,500
●CZ-8TM2	モデムユニット	（￥49,800）	▶特価￥38,000
●CZ-64H	増設ハードディスク	（￥120,000）	▶大特価
●CZ-6TU GY/BK	RGBシステムチューナー	（￥33,100）	▶特価￥25,000
●BF-68PRO	高性能CRTフィルター	（￥19,800）	▶特価￥15,500
●CZ-6MO1	光磁気ディスクユニット	（￥450,000）	▶特価￥333,000
●CZ-6BS1	SCSIインターフェースボード	（￥29,800）	▶特価￥22,200
●CZ-6BL2	LANボード	（￥298,000）	▶特価￥220,000
●CZ-6BV1	（ビデオボード）	（￥21,000）	▶特価￥15,500

特選周辺機器 （送料￥500）

● SX-68M MIDIインターフェースボード
（システムサコム）￥19,800…………特価￥14,800

● HGS-68（HAL研究所）
￥39,800…………特価￥26,300

■増設RAMボード＝I・Oデータ
①PIO-6BE1-A（1MB）
￥25,000…………特価￥17,800
②PIO-6BE2-2M（2MB）
￥50,000…………特価￥35,000
③PIO-6BE4-4M（4MB）
￥88,000…………特価￥62,000

■ビッグバーゲンセール実施中!! ゲームソフト（ビジネス）新製品続々入荷中!!

店頭ゲームソフトオール25%off! ビジネスソフト 25%より特価中

★通信販売お申込みのご案内★

〒144 東京都大田区蒲田4-6-7 TEL：03-3730-6271

お申込みはお電話でお願いします。お客様の〈住所〉〈氏名〉〈電話番号〉及び〈商品名〉をお知らせ下さい。● 入金確認後ただちに商品をご送付いたします。

現金一括払い
銀行振込：お近くの銀行より（電信扱い）にてお振込み下さい。
現金書留：封筒の中に住所・氏名・商品名をご記入の上当社までお送り下さい。

クレジット
専用お申込用紙をお送り致しますので、必要事項をご記入、ご捺印の上ご返送下さい。手続きは簡単です。

オクト ラクラク クレジット表

2回	3.5%	6回	4.5%	10回	6%	12回	6%
18回	11%	20回	12%	24回	12.5%	36回	17.5%
48回	23%	60回	29.5%				

振込先
富士銀行 久ヶ原（クガハラ）支店 ㊟No.1824
三菱銀行 蒲田支店 ㊟No.0278691
株式会社 億人（オクト）

※掲載の価格は変動しますので、まずは、お電話にてご確認ください。
※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは電話でお問合せ下さい。
※銀行振込、または、現金書留でご注文の際には、あらかじめ電話でご確認の上、お申し込み下さい。

超低金利クレジット

注目!! 夏のボーナス一括払い 手数料(金利)無料

(平成3年7月末をご利用下さい)

またまた 秋葉原で おなじみ

3/15〜4/14

●お近くの方
●本体単品で
●ビジネスソフ

HARD DISK UNIT(X68000専用)
アイテック(SCSI)
- ITX-80S(80MB/20ms)·定価¥128,000▶特価¥ 95,000
- ITX-130S(130MB/20ms)…定価¥158,000▶特価¥117,000

Fine Scanner-X68
(HAL研究所)X68000専用
■HGS-68(定価¥39,800)
特価¥26,500
(送料・消費税込み¥27,810)

X68000シリーズ専用 特価¥14,800
MIDIインターフェースボード
SX-68M(サコム)
(純生コンパチ)定価¥19,800
(送料・消費税込み¥15,759)

X68000メモリボード(シャープ&I/O・DATA)(送料¥500)

① CZ-6BE1(600C用) 定価¥35,000 (送料・消費税込¥27,295)……¥26,000
② PIO-6BE1-A 定価¥25,000 (送料・消費税込¥19,055)……¥18,000
③ PIO-6BE2-2M 定価¥50,000 (送料・消費税込¥36,565)……¥35,000
④ PIO-6BE4-4M 定価¥88,000 (送料・消費税込¥64,375)……¥62,000

ジョイスティック 送料¥500
- X-1PRO 定価¥9,500▶特価¥7,
- ASCII STICK 定価¥6,800▶特価¥5,

P&A超低金利クレジットをご利用ください!!

NEW X68000 SUPER/SUPER-HD/PROII/PROII-HD (送料・消費税込)

SUPER
セットでお買い上げの方に●ディスケット10枚 ●ジョイカード2ケ プレゼント中!!

NEW

Ⓐセット:CZ-604C-TN+CZ-606D-TN……定価¥427,800▶特価 価格はTEL下さい。

12回	?	24回	?	36回	?	48回	?	60回	?

Ⓑセット:CZ-604C-TN+CZ-613D-TN……定価¥483,000▶特価 価格はTEL下さい。

12回	?	24回	?	36回	?	48回	?	60回	?

SUPER-HD
セットでお買い上げの方に●ディスケット10枚 ●ジョイカード2ケ プレゼント中!!

Ⓐセット:CZ-623C-TN+CZ-606D-TN……定価¥577,800▶特価 価格はTEL下さい。

12回	36,300	24回	19,300	36回	13,400	48回	10,500	60回	8,900

Ⓑセット:CZ-623C-TN+CZ-613D-TN……定価¥633,000▶特価 価格はTEL下さい。

12回	39,700	24回	21,000	36回	14,600	48回	11,500	60回	9,700

セットでお買い上げの方に
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

先着100名様
セットでお買い上げの方には、ゲームソフト(V-BALL¥7,900)を、プレゼント!!
お早目に!!

PROII
セットでお買い上げの方に
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

Ⓐセット:CZ-653C+CZ-606D……定価¥364,800▶特価 価格はTEL下さい。

12回	21,800	24回	11,500	36回	8,000	48回	6,300	60回	5,300

Ⓑセット:CZ-653C+CZ-605D……定価¥400,000▶特価 価格はTEL下さい。

12回	24,200	24回	12,800	36回	8,900	48回	7,000	60回	5,900

Ⓒセット:CZ-653C+CZ-604D……定価¥798,000▶特価 価格はTEL下さい。
Ⓓセット:CZ-653C+CZ-613D……定価¥420,000▶特価 価格はTEL下さい。
Ⓔセット:CZ-653C+CU-21HD……定価¥433,000▶特価 価格はTEL下さい。

PROII-HD
セットでお買い上げの方に
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

Ⓐセット:CZ-663C+CZ-606D……定価¥474,800▶特価 価格はTEL下さい。

12回	30,200	24回	16,000	36回	11,200	48回	8,800	60回	7,400

Ⓑセット:CZ-663C+CZ-605D……定価¥510,000▶特価 価格はTEL下さい。

12回	32,300	24回	17,100	36回	11,900	48回	9,300	60回	7,900

Ⓒセット:CZ-663C+CZ-604D……定価¥489,800▶特価 価格はTEL下さい。
Ⓓセット:CZ-663C+CZ-613D……定価¥530,000▶特価 価格はTEL下さい。
Ⓔセット:CZ-663C+CU-21HD……定価¥543,000▶特価 価格はTEL下さい。

X68000シリーズ 〜P&Aスペシャルセット=限定誌上販売!!

台数限定 送料、消費税込み ※セットでお買い上げの方に、●ディスケット10枚、●ジョイカード2個 プレゼント中!!

EXPERTII

Ⓐセット:P&A厳選セット
■CZ-603C
(本体価格¥338,000)
+
■CZ-606D
(モニター定価¥79,800)
▶P&A超特価 ¥304,000

Ⓑセット
■CZ-603C+CZ-604D
定価¥432,800…▶特価¥309,000

Ⓒセット:
■CZ-603C+DZ-605D
定価¥453,000…▶特価¥322,000

Ⓓセット:
■CZ-603C+CZ-613D
定価¥473,000…▶特価¥342,000

Ⓔセット:
■CZ-603C+CU-21HD
定価¥486,000…▶特価¥347,000

EXPERT-HD

Ⓐセット:P&A厳選セット
■CZ-612C(ブラック)
(本体価格¥466,000)
+
■CZ-606D(ブラック)
(モニター定価¥79,800)
▶P&A超特価 ¥335,000

Ⓑセット:
■CZ-612C+CZ-604D
定価¥560,800…▶特価¥340,000

Ⓒセット:
■CZ-612C+CZ-605D
定価¥581,000…▶超特価¥359,000

Ⓓセット:
■CZ-612C+CZ-613D
定価¥601,000…▶超特価¥372,000

Ⓔセット:
■CZ-612C+CU-21HD
定価¥614,000…▶超特価¥386,000

■NEC=モデム(定価¥44,800)
⊙COMSTARZ 2424/5
●2400/1200bps全二重
●MNP5クラス
●インターフェース付
P&A超特価
¥25,000
(送料・消費税込み¥26,780)

■ALL in Note
フリートップ
パーソナルコンピュータ
⊙AX-286 N-H2
(定価¥398,000)
P&A超特価
価格はTEL下さい。

●本広告の掲載の商品の価格については、消費税は含まれておりません。 ●営業時間=平日AM10:00〜PM7:00、日祭AM10:00〜PM6:00

~84回払いまでOK!!

★頭金なし! ★即日発送

P&Aがズバリ超特価セールでご奉仕!!

●価格は流通事情により変動致しますので、銀行振込・書留等の送付前に、あらかじめお電話にてご確認下さい。

寄り下さい。専門係員が説明いたします。
で受付します。詳しくは電話にてお問合せ下さい。
の20%引きOK! TELください。

全国通販

超特価でクレジットが組める!!

X68000用ソフトコーナー(送料1ヶ~5ヶまで¥500)

商品	定価	特価
Z's STAFF PRO68K Ver.2.0(ツァイト)	¥58,000	¥39,500
Z's TRIPHONY デジタルクラフト(ツァイト)	¥39,800	¥27,800
テラッツォ(ハミングバード)	¥19,400	¥14,200
KAMIKAZE(サムシング・グッド)	¥68,000	¥44,800
C &.Professional Pack(マイクロウェアジャパン)	¥58,000	¥43,400
Finai Ver3.2(エーエスピー)	¥38,000	¥29,600
C-compiler PRO68K Ver.2 CZ-245L	¥44,800	¥33,300
CARD PRO68K CZ226BS	¥29,800	¥21,200
XBAS to C CHECKER CZ-260LS	¥9,800	¥7,400
OS-9/X68000 CZ219SS	¥29,800	¥22,500
AI-68K CZ234LS	¥188,000	¥138,000
THE 福袋 V2.0 CZ224LS	¥9,900	¥7,400
SOUND PRO68K CZ-214MS	¥15,800	¥11,400
MUSIC PRO68K CZ213MS	¥18,800	¥13,400
Sampling PRO68K CD215MS	¥17,800	¥12,700
MUSIC-studio PRO68K CZ-252MS	¥15,800	¥21,400
MUSIC-PRO68K〔MIDI〕247MS	¥28,800	¥20,700
New-print Shop 221HS	¥19,800	¥15,500
Communication 223CS	¥19,800	¥14,200
Communication Ver.2 CZ-257CS	¥19,800	¥15,500
C-TRACE68 Ver.3.0(キャスト)	¥98,000	¥74,600
サイクロンEXPRESS α68	¥98,000	¥69,800
G68K Ver2 PRO	¥22,000	¥17,500
SX-WINDOW CZ-259SS	¥6,800	¥4,900
Gツール(ザインソフト)	¥28,000	¥18,900
たーみのる2(SPS)	¥17,800	¥13,300
マジックパレット(ミュージカルプラン)	¥19,800	¥14,500
Hyper word CZ-251BS	¥39,800	¥29,600

●ゲームソフト20%OFF OK!!(一部ソフト除く)

X68000用ハードディスク(送料¥1,000)

アイテム

商品	定価	特価
HXD-040(40MB/23ms)	¥118,000	¥88,000
HXD-042(増設用)	¥128,000	¥95,000

アイテック

商品	定価	特価
ITX-640(40MB/28ms)	¥158,000	¥83,000
ITX-680(80MB/20ms)	¥198,000	¥97,000

プリンター(ケーブル・用紙付)(送料¥1,000)

商品	定価	特価
CZ-8PC5-BK NEW	¥96,800	¥72,000
CZ-8PK10	¥97,800	¥73,000
CZ-8PG2	¥160,000	特価価格はTEL!!
CZ-8PG1	¥130,000	特価価格はTEL!!

モデムコーナー(送料¥1,000)

商品	定価	特価
Ⓐ MD-24FS5(オムロン)	¥49,800	¥31,500
Ⓑ MD-24FS7(オムロン)	¥64,800	¥43,500
Ⓒ コムスター2424/4(NEC)	¥38,800	¥25,900
Ⓓ コムスター2424/5(NEC)	¥44,800	¥27,500

P&A特選パソコンラック(送料無料)移動自由(キャスター付)

周辺機器コーナー(送料¥500)

商品	定価	特価
Ⓐ CZ-8NSI	¥188,000	¥145,000
Ⓑ CZ-6VTI	¥69,800	¥52,500
Ⓒ CZ-6TU	¥33,100	¥24,500
Ⓓ BF-68PRO	¥19,800	¥15,300
Ⓔ CZ-6BEI	¥35,000	¥26,000
Ⓕ CZ-6BEIA	¥38,000	¥28,600
Ⓖ CZ-6BE2	¥79,800	¥60,000
Ⓗ CZ-6BE4	¥138,000	¥103,000
Ⓘ CZ-6BFI	¥49,800	¥38,200
Ⓙ CZ-6BPI	¥79,800	¥60,000
Ⓚ CZ-6BMI	¥26,800	¥20,300
Ⓛ CZ-6EBI	¥88,000	¥66,500
Ⓜ AN-S100	¥36,600	¥28,500
Ⓝ CZ-6SDI	¥44,800	¥35,000
Ⓞ CZ-6BN1	¥29,800	¥22,600
Ⓟ CZ-6BV1	¥21,000	¥15,900
Ⓠ CZ-64H	¥120,000	¥91,500
Ⓡ CZ-6BG1	¥59,800	¥45,000
Ⓢ CZ-6BU1	¥39,800	¥30,300
Ⓣ CZ-6PVI	¥198,000	¥153,000
Ⓤ IO-735X	¥248,000	¥190,000
Ⓥ CZ-6BS1	¥29,800	¥22,300
Ⓦ CZ-8NJ2	¥23,800	¥18,500
Ⓧ CZ-6BL2	¥298,000	¥222,000

中古パソコン(セットはモニター付)送料¥2,000

商品	価格
X68000セット	¥180,000
X68000 ACEセット	¥200,000
X68000 ACE-HDセット	¥215,000
EXPERTセット	¥230,000
EXPERT-HDセット	¥265,000
PROセット	¥250,000
X68000PRO-HDセット	¥270,000
EXPERTⅡセット	¥250,000
EXPERTⅡ-HDセット	¥320,000
PROⅡセット	¥240,000
PROⅡ-HDセット	¥310,000

中古パソコンはP&Aにおまかせ!!

その場で高価現金買取り・高価下取りOK!!

■まずはお電話下さい。
03-651-1884 FAX:03-651-0141

■下取り・買取りでお急ぎの方、直接当社に来店、または、宅急便にてお送り下さい。

●下取りの場合……価格は常に変動していますので査定額をお電話で確認して下さい。(差額は、P&A超低金利クレジットをご利用下さい。)

●買取りの場合……現品が着き次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。

●近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、¥1,000,000までお支払い致します。

《便利な超低金利クレジットをご利用下さい》

●月々¥1,000円からOK!! ●ボーナス払いOK(夏冬10回までOK)
●支払い回数 1回~84回 ●お支払いは、8ヶ月先からでもOK!!

アフターサービス万全
全商品保証付。専門の担当者がお客様の立場で対応します。
初期不良、輸送トラブル etc.
万が一初期不良、輸送トラブルが発生しました際には、即交換させていただきます。

●定休日/毎週水曜日=第3水曜・木曜は連休とさせていただきます(祭日の場合は翌日になります)

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。(プリンター・フロッピーの場合、本体使用機種名を明記のこと)

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。
(電信扱いでお振込み下さい。)

〔振込先〕住友銀行 新小岩支店
当No.263914 (株)ピー・アンド・エー

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
●1回~84回払いまで出来ます。但し、1回のお支払い額は¥1000円以上。

超低金利クレジット率

回数	3	6	10	12	18	24	36	48	60	72	84
手数料	3.5	4.5	6.0	6.0	11.0	12.5	17.5	23.0	29.5	38.0	45.5

南口 徒歩1分
至秋葉原 JR新小岩駅 バスターミナル 住友BK 東海BK リブ 北海道拓殖BK P&A本店 サンチェーン 蔵前通り 至千葉

●マイコン
●ビデオ
●ビデオテープ

P&A

株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目1番地19号
☎03-3651-0148(代) FAX.03-3651-0141

営業時間
平日:AM10:00~PM7:00
日祭:AM10:00~PM6:00

●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

ワールドインアオヤマのコンセプトショップ 池袋本店にCOME ON!

● TOKYU HANDS となり AMLUX 前

X68000 PRO II

X68000EXPERT II X68000 新L コース

CZ-653C(本体)……¥285,000
CZ-6050……¥115,000
住友3M5'2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

合計 ¥418,000 ➡ 現金特価
安すぎて表示できません。
クレジットでもお申し込み出来ます。

X68000PRO II X68000 新J(ジェイ) コース

CZ-604C TN(本体)……¥348,000
CZ-606TD(0.31カラーディスプレー)……¥79,800
CZ-8NJ2(インテリジェントスティック)……¥23,800
住友3M5'2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

定価合計 ¥474,600➡現金大特価
安すぎて表示できません。
クレジットでもお申し込み出来ます。

X68000PRO II X68000 G コース

CZ-653C(本体)……¥285,000
CZ-606D(0.31カラーディスプレー)……¥79,800
住友3M 5'2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

定価合計 ¥387,800➡¥249,800

¥6,900×48回	ボなし 頭なし
¥12,600×24回	ボなし 頭なし

X68000PRO II X68000 新P(ピー) コース

CZ-652CBK(本体)……¥298,000
CZ-602D BK(0.39カラーディスプレーテレビ)¥99,800
住友3M5'2HDブランクディスケット……¥18,000
御希望ゲームソフト(人気ソフト上記よりお選び下さい)¥サービス

定価合計 ¥415,800➡¥257,000
安すぎて表示できません。
クレジットでもお申し込み出来ます。

全国出張サポート★
私共にてご購入いただいた X68000 は全国出張サポートが受けられます。

X68000 EXPERT II

X68000EXPERT II X68000 新K コース

CZ-653C(本体)……¥285,000
CZ-604D(0.31ステレオスピーカ付ディスプレー)……¥94,800
御希望ゲームソフト……サービス

合計 ¥379,800 ➡ 現金特価
安すぎて表示できません
クレジットでもお申し込み出来ます

X68000EXPERT II X68000 新D コース

CZ-603C(本体)……¥338,000
CZ-604D(0.31ステレオスピーカ付ディスプレー)……¥94,800
住友3M5'2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

合計 ¥450,800 ➡ 現金特価
安すぎて表示できません。
クレジットでもお申し込み出来ます。

X68000EXPERT II X68000 新C コース

CZ-603C(本体)……¥338,000
CZ-605D(0.39ステレオスピーカ付テレビ)……¥115,000
住友3M5'2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

合計 ¥475,000 ➡ 現金特価
安すぎて表示できません
クレジットでもお申し込み出来ます

X68000EXPERT II X68000 A コース

CZ-603C(本体)……¥338,000
CZ-606D(0.31カラーディスプレー)……¥79,800
住友3M 5'2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

定価合計 ¥440,800➡¥305,000
安すぎて表示できません。
クレジットでもお申し込み出来ます。

X68000お買上げのお客様へ
各コースで御希望ソフトは「サンダーブレード」「ダウンタウン熱血物語」「ニュージーランドストーリー」「沙羅曼蛇」「ツインビー」「フルスロットル」「パックマニア」「ビーチバレー」「アルカノイド」「熱血高校ドッチボール」のうちいずれかからお選び下さい

SUPER HD

X-68000 SUPER HD X68000 O(オー) コース

CZ-623CTN(本体80M HDD内蔵)……¥498,000
CZ-606DTN(0.31カラーディスプレーチタン)¥79,800
住友3M 2HDブランクディスケット……¥18,000
御希望ゲームソフト……サービス

定価合計 ¥600,800➡現金大特価
安すぎて表示できません
クレジットでもお申し込み出来ます

SHARP X68000 パーツセット

最新お買得ソフトセット X68000 パーツセット1

CZ-652CBK(本体)……¥298,000
CZ-606DBK(0.31カラーディスプレー)……¥84,800
住友3M5'2HDブランクディスケット……¥18,000
御希望ゲームソフト(人気ソフト上記よりお選び下さい)……¥サービス

定価合計 ¥400,800➡¥245,000
安すぎて表示できません。
クレジットでもお申し込み出来ます。

ステレオMIDI音源セット X68000 パーツセット2

CZ-6BM1(MIDIボード)……¥26,800
CM-32L(ローランドMIDI音源)……¥69,000
MA-12AV×2(ボースアンプ内蔵スピーカー)¥28,000
MUSIC-PRO-MIDI(ソフト音源)……¥28,800
ソングライブラリ(ソフト音源データ)……¥8,800

定価合計 ¥161,400➡¥134,900
安すぎて表示できません。
クレジットでもお申し込み出来ます。

ステレオMIDI音源セット・オールブラック X68000 パーツセット3

SX-68M(MIDIボード)……¥19,800
MT-32(ローランドMIDI音源)……¥64,000
MA-12AV×2(アンプ内蔵スピーカー)……¥28,000
CF-10(コンピュータミキサー)……¥22,000
PC-200(キーボード)……¥36,000
MU-1(ソフト音源)……¥19,800
ソングライブラリ(ソフト音源データ)……¥8,800
—お買上げのX68000と合せて御利用いただけます。—

定価合計 ¥198,400➡¥163,000
安すぎて表示できません。
クレジットでもお申し込み出来ます。

カラーグラフィックセット X68000 パーツセット4

GT-1000(カラーハンディスキャナ)……¥79,800
≠5220(接続ケーブル)……¥7,500
Z's STAFF・PRO(グラフィック作成ソフト)……¥58,000

定価合計 ¥145,300➡¥103,800
安すぎて表示できません。
クレジットでもお申し込み出来ます。

MUSIC PRO-68K ¥18,800➡現金特価、OS-9 X68000 ¥29,800➡¥22,300 Human68K ver2.0 ¥9,800➡現金特価 たーみのるII ¥12,800➡現金特価 Z5 STAFF PRO-68K ¥19,800➡現金特価 C-TRACE ¥68,000➡¥38,000 Kamikaze ¥68,000➡¥49,600 Canvas PRO 68K(ドローグラフィックツール)¥29,800➡現金特価

高解像度カラーで金や銀まで表現 X68000

CZ-604C(本体)……
CZ-605D(0.39ステレオディスプレーテレ…
住友3M2HDブランクディスケット…
御希望ゲームソフト……

定価合計 ¥481,000➡現…
安すぎて表示できま…
クレジットでもお申し込み…

SHAR…

グレー限定お買徳セット X68000

CZ-652CGY(本体)……
CZ-604DGY(0.31ステレオスピーカー付ディス…
住友3M2HDブランクディスケット…
御希望ゲームソフト……

定価合計 ¥410,800➡現…
安すぎて表示できま…
クレジットでもお申し込み…

チタンカラーも入ってますます充実 X68000

CZ-604CTN(本体)……
CZ-606DTN(0.31カラーディスプレー…
住友3M 5'2HDブランクディスケ…
御希望ゲームソフト……

定価合計 ¥400,800➡現…
安すぎて表示できま…
クレジットでもお申し込み…

■買ったお客様でしかわからないこのサービス!——ぜったいX68000を買うならアオヤマがオトク
●以前当社にてX68000及びX-1を御購入いただいたお客様に限り、CZ-8PC5〔定94,800〕を大特価にてお届けいたします。会員の方は会員ダイアルにてCall!
●X68000をセットでお買い上げいただいたお客様に限り、アスキーターボステックを特価¥2,980、XE-1PROを特価¥6,700また CTRACEを特価¥39,800にてCZ-8NJ2〔インテリジェントコントローラ〕〔¥23,800〕を特価¥16,900にてお届けいたします。御注文の際に合わせてお申し込み下さい。

X68000をはじめソフト&周辺機器類は、当社池袋店・札幌店・旭川店・福岡店にて実演中です。各店X68000コーナーが常設されております。

X68000ソフト&周辺機器

品名	価格	品名	価格	品名	価格
SCSIボード(CZ-6BSI)	¥29,800➡現金特価	BF-68PRO	¥19,800➡¥16,800	Communication PRO-68K	¥19,800➡現金特価
システムサコムMIDIボード(SX-68M)	¥19,800➡¥15,300	CZ-6TU	¥33,100➡現金特価	Stationary PRO-68K	¥14,800➡現金特価
LANボード	¥268,000➡¥201,000	オムロンMD-24FP4 II	¥38,800➡現金特価	DATA PRO-68K	¥58,000➡¥43,500
RS-232Cケーブル(平行)	¥7,200➡現金特価	オムロンMD-24FP5 II	¥42,800➡¥33,000	BUSINESS PRO-68K	¥68,000➡¥51,000
RS-232Cケーブル(クロス)	¥7,200➡現金特価	ローランドMT-32	¥64,000➡¥54,400	NEW Printshop PRO-68K	¥19,800➡現金特価
インテリジェントコントローラ	¥23,800➡¥18,900	Hyperword	¥39,800➡現金特価	グラフィックライブラリ vol.1	¥8,800➡現金特価
トラックボール	¥13,800➡¥12,000	CYBERNOTE PRO68K	¥19,800➡現金特価	グラフィックライブラリ vol.2	¥8,800➡現金特価
ジョイカード(延長コード付)	¥3,200➡¥2,900	C compiler PRO-68K	¥44,800➡¥33,600	Musicstudio PRO-68K ver1.1	¥28,800➡¥21,600
CZ-8BS1(X-1用)	¥23,800➡現金特価	CARD PRO-68K	¥29,800➡現金特価	MUSIC PRO-68K (MIDI)	¥28,800➡現金特価
拡張I/Oボックス	¥88,000➡現金特価	CARD PRO システム手帳リフィル集	¥9,800➡現金特価	ソングライブラリ 101曲集	¥8,800➡現金特価
アンプ内蔵スピーカーシステム	¥36,600➡現金特価	CARD PRO 活用フォーム集	¥9,800➡現金特価	Sampling PRO-68K	¥17,800➡現金特価
システムラック	¥44,800➡¥35,800	SX-WINDOW ver1.0	¥6,800➡現金特価	SOUND PRO-68K	¥15,800➡現金特価

X68000シリーズ周辺機器

品名	価格	品名	価格	品名	価格
CZ-6BS1	¥188,000➡¥141,000	CZ-8PC5	¥94,800➡現金特価	I/Oデータ 2MB増設RAM	¥50,000➡¥36,500
CZ-6BN1	¥29,800➡現金特価	IO-735X	¥248,000➡現金特価	I/Oデータ 4MB増設RAM	¥88,000➡¥64,000
CZ-6VT1	¥69,800➡¥52,400	CZ-8PK10	¥97,800➡現金特価	GP-IBボード	¥59,800➡現金特価
CZ-6BV1	¥21,000➡現金特価	1MB増設RAM(CZ-600C専用)	¥35,000➡¥28,000	増設用RS-232Cボード	¥49,800➡現金特価
CZ-6PV1	¥198,000➡¥148,500	1MB増設RAM	¥28,000➡¥22,400	ユニバーサルI/Oボード	¥39,800➡現金特価
CZ-8PC3	¥65,800➡¥39,000	2MB増設RAM	¥79,800➡現金特価	数値演算プロセッサ	¥79,800➡現金特価
CZ-8PG1	¥130,000➡¥97,500	4MB増設RAM	¥138,000➡現金特価	FAXボード	¥79,800➡¥55,800
CZ-8PG2	¥160,000➡現金特価	I/Oデータ 1MB増設RAM	¥25,000➡¥18,000	MIDIボード	¥26,800➡現金特価

X68000万全のサポート AOYAMAにて購入のX68000は万一故障の場合でも全国どこでも出張サービスがうかがいます。万一の場合ワールドインアオヤマサポート係にお電話下さい。お客様のお名前と電話番号だけで手続きは完了。

組合せ自由	激安金利にキャンパスクレジット	ゆっくり、お支払いは8ヵ月先から
各コース以外の組合せもコースをベースに周辺を合せたセット……お支払いだって御希望のパターンをお組みいたします さあ、ご相談もお見積りも受注センターもしくは各店へお気軽に	手続きカンタン、大学生の為の超低金利クレジット。20歳以上の学生の方は原則として保証人様には連絡いたしません。	クレジット業界最低の金利を有効に使って、支払いは最長8ヵ月後から始まるクレジットでも。

AOYAMAだからオトク

パソコン通信セット〔ソフトたーみのる2(¥12,800)+モデムMD24FP5II 2400ボモデム(¥42,800)〕➡¥42,800 通信セット〔ソフトX Talk-68K(¥12,800)+モデムMD12FS1200ボモデム(¥21,000)〕➡¥27,300 X68000接続電子手帳セット〔ケーブルCE-300L(¥3,500)+サイバーノート68K(¥19,800)+電子手帳PA-8500(¥28,000)〕➡¥38,600 ★CU-21HD〔ステレオスピーカ付21インチディスプレー〕……¥148,000➡現金大特価 ★CZ-604D〔ステレオスピーカ付603Dディスプレー〕……¥93,000➡現金大特価

ワールドインアオヤマの9ハートサービス

実績のメンテナンス/完全バックアップシステム

●パソコンのセオリーが体験できる。●パソコンだけのスペース。

池袋本店にどうぞ!

営業時間11:00AM〜19:00PM
東京都豊島区東池袋1-28-1

Computer PLAZA
CREATIVE LIFE STORE TOKYU HANDS
東急ハンズ池袋店となり
AMLUX 前

ワールドインアオヤマにおまかせ下さい。

—— クレジット ——
業界一番のスーパークレジットで。
学生の味方、キャンパスクレジットがますますワイドに。
お支払いはナント!84回まで。
ゆうゆうお支払いは8ヶ月先から。
—— サポート ——
万一のときでも完全バックアップ。
—— 会員・保険 ——
最先端の"CLUB246"。
インシュアランス・サービス
グーンとお得な下取りシステム。
—— ショップ ——
全店統一のサービス 見て、触れて、納得。
BEST SHOPS BEST SERVISE。
—— サービス&ポリシー ——
グーンとお得な下取りシステム。

〔返品・交換について〕

返品・交換は商品到着後3日以内にお願いいたします。

〈返品・交換をお受けする場合〉
●不良品(キズ・汚損)が届いた場合。
●ご注文商品と違う商品が届いた場合
上記の場合、返送料は当社が負担いたします。ただしお客様の事由(注文ミスなどによる返品の場合、返送料はお客様の負担となりますのでご了承下さい。
〈次の場合返品・交換はお受けできませんのでご注意下さい。〉
①1度ご使用になった商品。②お客様の責任でキズや汚れが生じた商品
③全ソフト商品

〔日中ご不在がちな方はお振込〕

昼間ご不在がちの方、お急ぎの方は銀行振込が便利です。
お申し込みからお届けまで、お時間をとりません。
●銀行振込みの場合

取引銀行	住友銀行 池袋支店
口座番号	普通 1065392
口座名	株式会社 ワールドイン アオヤマ

返品・交換についてのお問い合わせは
お客様相談室 03-3987-7795

〔電話でのご注文の場合〕

平成3年1月1日から東京は局番が4桁になり頭に3がつきます

03-3987-7771

お電話番号はおかけ間違いのないようにお願いします。
北海道受注センター 011-251-6771
九州受注センター 092-672-7771
お好きな時間にお電話を!

テレフォンショッピング

電話受付時間 ■月曜日〜金曜日 10:30〜20:30 ■土・日曜日・祭日 10:30〜19:00

〔クレジットカードもOK!〕

クレジットカードをお持ちの方お支払は1回払いです。お申し込みの際、①カード名②会員No.③有効期限をご連絡下さい。
カードでお申し込みの場合、販売価格が変わりますのでお電話にてお問い合せ下さい

サポート

〈万一のときにも完全バックアップ〉
万一の初期不良があった場合でも当社では万全の体制でお客様をフォロー致します。

〔クレジットご利用の…

手続きはお電話で…
お電話をいただきましたら…き落とし口座とお客様が…いたことがすべて解消…トスタッフがお役に立ちま…
※18歳未満の方はご両親のいずれ…

業界一番のスーパー…
ご都合に合わせて自由に…金額を設定できます
手続きはとても簡単、そして…もちろん手数料は超低金…お支払いも月々1,000円か…お得に使いやすくなったア…ジットです

学生の味方キャンパス…
キャンパスクレジットのワク…がりました 最高10万円ま…でき、手続きもその場で即…も月々1,000円からと、とっ…アオヤマは皆さんを応援し…

お支払いはナント8…
お支払いは少なく、ゆっくりと…オヤマなら最長84回まで自由…ボーナス額を増やしたり減らし…ん、ボーナス一括や二括払い…

ゆうゆうお支払いは…
欲しい機種は今すぐ欲しい…だけど以前のローンが残っ…就職してから始めたい…という方にお勧め
8ヶ月先までならいつ始め…これならとっても安心ですね

●配送日指定OK ●土・日配送OK ●札幌PAXに買取専門ショップOPEN ●専門スタッフが常時待機 ●中古パソコン4ヶ月間保証 ●FAX見積りサービス ●もちろんクレジットカードも使えます ●カード払いでキャッシュレス

お客様相談室=03-3987-7795 すでにご注文いただいているお届け時間(時期)やメンテナンス、その他のお問い合わせは右記へお電話下さい。

旭川店
旭川市4条8丁目ツジビル
■営業時間/11:00〜19:00

札幌店
札幌市中央区南2条西3丁目
リンクエギビル3F
■営業時間/11:00〜19:30

札幌PAX店
WORLD IN AOYAMA
札幌市中央区南2条西2丁目
ブロックビル6F
■営業時間/11:00〜19:30

池袋店ソフト店
豊島区東池袋1-28-6
パールシティビル2F
■営業時間/11:00〜19:00

池袋ショールーム
豊島区東池袋1-27-12明治生命池袋ビル9F・首都高速道路側
■営業時間 11:00〜19:00

福岡店
福岡市中央区渡辺通り4-9…
ユーテックプラザ4F・地下鉄天…
■営業時間 11:00〜19…

システムソフトが広げる、面白くする、
X68000エキサイティング・シーン。

SystemSoft X68000series Line up

X68000シリーズ ■5"-2HD
アナログRGB(31KHz対応)ディスプレイをお使いください。
アタリ社仕様の2トリガージョイスティック、ジョイパッドが使用できます。
人以上でプレイする場合は上記のジョイスティック、ジョイパッドが必要です。

価格 ¥7,800

コウフンのバクハツだ。
愉快な爆弾アクションゲーム。

爆弾で敵キャラや対戦相手をぶっとばす人気のアクションゲーム「ボンバーマン」。そのオモシロ爆弾が、ついにX68000へも仕掛けられた。ひとりでだって充分すぎるほど楽しめるこのゲーム。でも、もっともっと熱くなりたい!という君には、最大4人が同時に遊べる〈バトルゲーム〉がイチオシ。何をしでかすか予測不能の人間相手に、爆弾を仕掛け合いながら生き残りを競う超興奮のサバイバル。スリルと緊張感の連続に、ボタンを押す手にも力が入り、性格マル出しのプレイも思わずポロリ。これはもう、オキテ破りの爆弾デスマッチ。さぁ仲間を集めて、栄光のチャンプを目指し、時を忘れるバトルパーティの始まりだ。

※画面は開発中のものです。

3月発売予定

キャンペーン版
大戦略II
CAMPAIGN VERSION

X68000シリーズ ■5"-2HD(2枚組)
アナログRGB(31KHz対応)ディスプレイをお使いください。

価格 ¥9,800

戦略は、
いま新たなる次元へ。

つねに未知なる戦略シーンを追い求める勇者たちへ、新たな興奮と感動を贈ろう。ストラテジック・シミュレーションの頂点に立つ不朽の名作「大戦略シリーズ」が、ついにX68000に登場。「キャンペーン版大戦略II」――通常のマップモードに加え、戦闘経験をへて熟練度の上がったユニットをそのまま引き継いで次のステージに進み、8ステージの連続制覇に挑む<キャンペーンモード>を導入。都市に、孤島に、そして平原に展開する熾烈な戦いの舞台を、じっくり楽しむことができる。ウインドウメニュー、マウス対応など操作性も一段とアップ。いまX68000の広大なフィールドで、戦略は確かな進化を迎えるだろう。

※画面は開発中のものです。

発売日等の最新情報を下記のとおりテレフォンサービスにてご案内いたしております。どうぞお気軽にご利用ください。

新製品の発売日および内容のご案内は…
テレフォンサービス専用電話 東京:03-3326-8710
福岡:092-752-2602

商品のお申込みおよび発売日に関するお問い合わせは…
営業部専用電話 092-752-5262
土曜日、日曜日、祝祭日は営業いたしておりません。

商品に関する技術的なお問い合わせは…
ユーザーサポート専用電話 092-752-5278
月〜金 9:00〜12:00 13:00〜17:00(祝祭日を除く)

総合カタログをご希望の方は請求券をはがきに貼り、住所・氏名・年齢・電話番号・使用機種名を明記の上、弊社宛にご送付ください。
製品の仕様は、機能・性能の改善のため将来予告なしに変更することがあります。
表示価格に消費税は含まれておりません。

株式会社 システムソフト
〒810 福岡市中央区天神3丁目10-30

総合カタログVol.8請求券
Oh!X4月号
有効期限:1991年4月末

全国通販

SHARP 認定 PPO-SHOP **O.A.ランド**

(TEL) 03-3770-8855

■アフターサービス万全のサポート体制
●下取・買取は電話で見積りしております。責任を持って下取りさせて頂きます。
●ご注文、お問合せは…。午前10時から午後7時まで
●商品のお届けは…入金確認後、即日発送致します。
■TEL・FAXのお見積OK.!!
■低金利クレジットをご利用下さい。

▶3・15〜4・14

SHARPのことなら なんでもおまかせ!!

大徳買セール! 安く値切ってネ。(本体セット:送料 消費税込み)
お電話下さい。㊙価格をお知らせいたします。

流通事情により、広告表示価格は、お安くなる場合がありますので、ドンドンお電話下さい。

CYBER STICK
■CZ-8NJ2 (定価¥23,800) OAランド特価 ▶¥18,000

電子手帳 ●見やすい漢字4桁表示!! 情報任時代の必需品!!
■PA-9500(¥48,000)…▶特価¥38,000
■PA-8500(¥28,000)…▶特価¥15,000
■PA-7500(¥22,000)…▶特価¥12,000

2月中旬より、お店の場所が移転しました。装いも新たなオーエーランドです。ぜひお立寄り下さい。

SHARP X68000シリーズセット (送料・消費税込み)

X68000 SUPER NEW

①CZ-604C-TN+CZ-613D-TN
定価合計¥483,000

12回	31,100	24回	16,500
36回	11,500	48回	9,000

②CZ-604C-TN+CZ-606D-TN
定価合計¥427,800

12回	27,600	24回	14,600
36回	10,200	48回	8,000

■CZ-604C
特価~~¥248,000~~
■CZ-623C
特価~~¥498,000~~

X68000 SUPER-HD

①CZ-623C-TN+CZ-613D-TN
定価合計¥633,000

12回	40,600	24回	21,500
36回	15,000	48回	11,700

②CZ-623C-TN+CZ-606D-TN
定価合計¥577,800

12回	37,100	24回	19,700
36回	13,700	48回	10,700

X68000 PROII

①CZ-653C+CZ-613D
定価合計¥420,000

1回	TEL下さい	12回	25,400
24回	13,500	36回	9,400

②CZ-653C+CZ-605D
定価合計¥400,000

1回	TEL下さい	12回	24,100
24回	12,800	36回	8,900

③CZ-653C+CZ-606D
定価合計¥364,800

1回	TEL下さい	12回	21,800
24回	11,500	36回	8,000

■CZ-653C
特価~~¥285,000~~
■CZ-663C
特価~~¥395,000~~

X68000 PROII-HD

①CZ-663C+CZ-613D
定価合計¥530,000

1回	383,000	12回	35,000
24回	18,000	36回	12,900

②CZ-663C+CZ-605D
定価合計¥510,000

1回	370,000	12回	33,600
24回	17,800	36回	12,400

③CZ-663C+CZ-606D
定価合計¥474,800

1回	344,000	12回	31,300
24回	166,000	36回	11,500

上記組合せのディスプレイ(モニター)変更自由!!
詳しくは、お電話にてお問い合せ下さい!!

■期間中、セットでお買い上げの方には、①サンダーブレード(ゲームソフト ¥9,500)と②テトリスやドルアーガの塔などの入ったゲームパックをプレゼント!!

X68000用SCSIハードディスク!!

キャラベル
① AV-040SC+ケーブル ……特価¥79,000 (合計定価¥116,000)
② AV-090SC+ケーブル ……特価¥114,000 (合計定価¥172,000)
③ AV-130SC+ケーブル ……特価¥139,000 (合計定価¥212,000)
④ AV-200SC+ケーブル ……特価¥208,000 (合計定価¥314,000)
⑤ AV-250SC+ケーブル ……特価¥266,000 (合計定価¥402,000)

アイテック
① ITX-80S ……特価¥88,500 (定価¥128,000)
② ITX-130S ……特価¥108,000 (定価¥158,000)
※X68000 SUPER以外の機種では、SCSIボードが必要となります。
⊙ SCSIボード ……特価¥22,000
⊙ 光ディスク ……特価¥320,000

周辺機器

■光磁気ディスクユニット
●CZ-6MO1 (定価¥450,000)
特価TEL下さい!!

■SCSIボード
●CZ-6BS1 (定価¥29,800)
特価TEL下さい。

■ビデオボード
●CZ-6BV-1 (定価¥21,000)
特価¥15,600

周辺機器コーナー 電話で値切ろう。

プリンターセットコーナー

①CZ-8PC5 NEW 定価¥96,800
●48ドット ●熱転写カラー 漢字プリンター
大特価TEL下さい!!
②CZ-8PK10(24ピン漢字プリンター136桁)
定価¥97,800…特価¥71,000
③CZ-8PG1(24ピンカラー漢字プリンター80桁)
定価¥130,000…特価¥93,000
④CZ-8PG2(24ピンカラー漢字プリンター136桁)
定価¥160,000…特価¥114,000

OAランド特選品.!!

■IO-735X(定価¥248,000)
●カラーイメージ ジェットプリンター
特価¥177,000

《計測技研》増設メモリ&プロセッサ

●高速増設メモリと数値演算プロセッサが一つのボードになった!!
● KGB-X68 PRK-00(¥34,000)…特価¥26,000
● PRK-01(¥58,000)…特価¥43,500
● PRK-02(¥74,000)…特価¥55,500
● PRK-03(¥98,000) 特価¥73,500
● PRK-04(¥122,000) 特価¥91,500
● PRK-10(¥72,000) 特価¥54,000
● PRK-11(¥96,000)…特価¥72,000
● PRK-12(¥112,000)…特価¥84,000
● PRK-13(¥136,000)…特価¥102,000
■KGB-X68PRK-14(¥160,000)‥特価¥115,000
■MC6888 1RC16(¥38,000)……特価¥28,500

X68000用周辺機器コーナー

①CZ-6VT1(カラーイメージユニット) 定価¥69,800 ……特価¥52,500
②CZ-8NS1(カラーイメージスキャナー) 定価¥88,000 ……特価¥138,000
③CZ-6BM1(MIDIボード) 定価¥26,800 ……特価¥20,500
④CZ-8NJ2(インテリジェント・コントローラー) 定価¥23,800 ……特価¥18,000
⑤CZ-6TU(RGBシステムチューナー) 定価¥33,100 ……特価¥25,000
⑥CZ-64H(増設ハードディスク) 定価¥120,000 ……特価¥89,000
⑦CZ-6EB1(拡張I/Oボックス=4スロット) 定価¥88,000 ……特価¥66,000
⑧CZ-6BP1(数値演算プロセッサボード) 定価¥79,800 ……特価¥60,000

I・Oデータ増設RAMボード

■PIO-6BE1-A (1MB) 定価¥25,000 特価¥17,800
■PIO-6BE2-2M (2MB) 定価¥50,000 特価¥35,500
■PIO-6BE4-4M (4MB) 定価¥88,000 特価¥62,200

モデム

オムロン	MD-1200A III	¥14,500
	MD-24FP4 II	¥27,500
	MD-24FP5 II	¥29,800
	MD-24FN4	¥28,000
	MD-24FN5	¥31,300
	MD-24FJ4	¥31,300
	MD-24FJ5	¥34,500
	MD-24FS4	¥28,500
	MD-24FS5	¥34,500
アイワ	PV-A24VM5	¥32,500
	PV-M24	¥28,800
NEC	COMSTAR 2424/4	¥28,800
	COMSTAR 2424/5	¥33,500

OAランド今月の大目玉!!=超A級中古品

★CZ-623C+CZ-613D(2台)㊭……特価¥436,000
★CZ-604C+CZ-606D(4台)㊭‥特価¥292,000
★CZ-603C"GY(1台)……特価¥180,000
★CZ-603C-BK(2台)㊭……特価¥190,000
★CZ-8PK9(1台)㊭……特価¥38,000
★CZ-8NS1(2台)……特価¥90,000
★CZ-8PC4(3台)……特価¥45,000
★CZ-8PC5(2台)㊭……特価¥69,800
★IO-735X(1台)㊭……特価¥174,000
★CZ-6M01(2台)㊭……特価¥315,000
※(　　)内の数字は2月末現在の在庫数です。本体セット、単体でも受付けますので、お電話下さい。㊭は、メーカー保証付です。

OAランド推奨ソフト

■SX-WINDOW (次世代インテリジェントソフト) 定価¥6,800 特価¥5,100
■CZ-245LS (C-コンパイラII) 定価¥44,800 特価¥33,500
■CZ-260LS (X Bas to C CHECKER) 定価¥9,800 特価¥8,000
■CZ-249GS (CANVAS-PRO68K) 定価¥29,800 特価¥22,300
■CZ-255GS/256GS (ドローグラフィックライブラリ1/2) 定価¥8,800 特価¥7,000
■CZ-219SS (OS9/X68,000) 定価¥29,800 特価¥23,800

通信販売のご案内

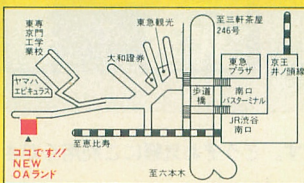
全国通販

■銀行振込で申し込みの方は商品名及びお客様の住所・氏名・電話番号をお知らせ下さい。
〔振込先〕第一勧業銀行 渋谷支店
普通No.1163457 ㈱オーエーランド

■年中無休です!!

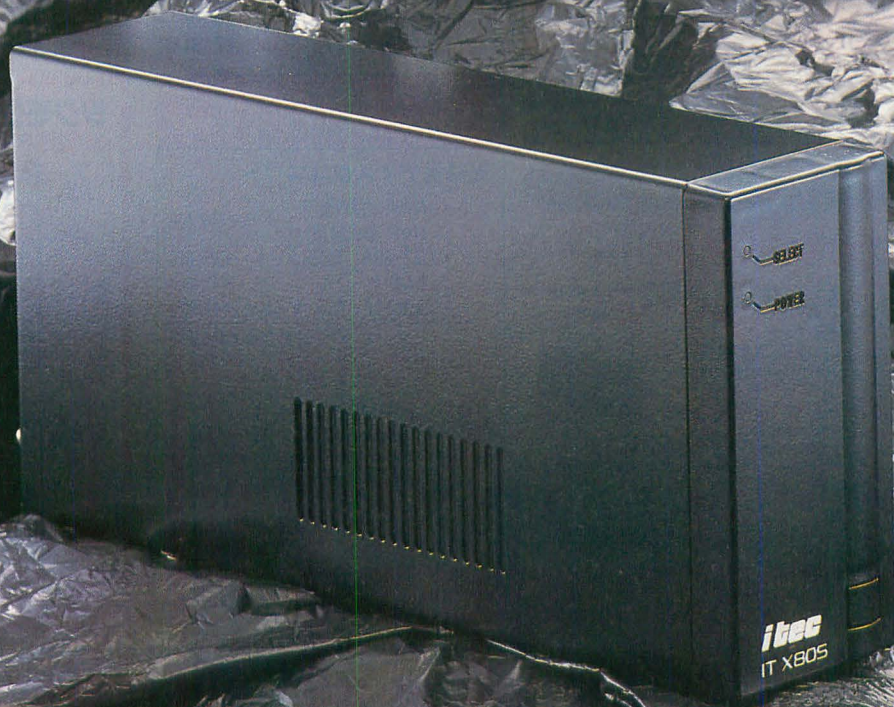
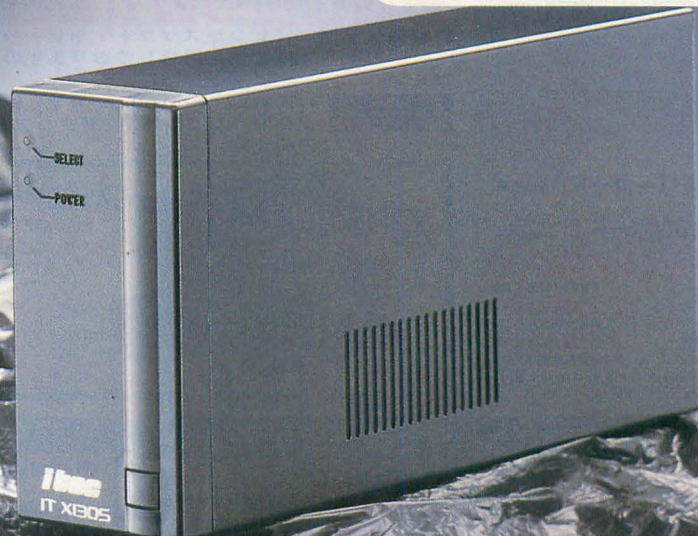
■現金書留で送金されるお客様は電話番号と商品名、数量を明記して同封して下さい。■クレジットでご購入を希望される方は申し込み用紙をお送り致しますのでご記入の上返送して下さい。20才以上の方は、原則として保証人不要です。クレジットは1〜60回払で月々5,000円よりご自由に設定できます。

クレジット表

3回	3.5%	6回	4.5%	10回	6%	12回	6%	15回	8.5%	18回	11%	20回	12%
24回	12.5%	30回	17%	36回	17.5%	42回	22.5%	48回	23%	54回	29%	60回	29.5%

㈱オー・エー・ランド
〒150 東京都渋谷区桜丘町3-13 アルカディア2F
☎(03)3770-8855

関東エリアの送料は、1個につき¥1,000です。 FAX(03)3770-7080
★全商品保証書付。専門のアドバイザーが、お客様のニーズに対応します。
★初期不良・輸送トラブル等に迅速に対応し、即交換させていただきます。

■表示価格は、税別表示です。詳しくは、お電話にて、お問い合せ下さい。掲載の価格は、2月下現在です。

'91 NEW WAVE
itec
HARD DISK UNIT
SHARP X68000専用
ックのハードディスクがさらに進化しました。
RP X68000専用SCSIハードディスクユニットIT X80S
X130Sの登場です。
ックでは早くからX68000専用ハードディスクに着手し今ま
ASIタイプハードディスクに多大なる評価を得てきました。
術をさらに進化させた結果次代が求めるSCSIタイプハー
スクの発表となったわけです。
0 SUPER HDの発売によりSCSIのもつ拡張性能の良
らためて認識されるようになりました。また従来のX68000
ズにもSHARP製SCSIインターフェースボードCZ-6BS1
ることにより完全にSCSI機器を周辺装置として接続が可
りました。アイテックのSCSIハードディスクはこのどち
対応できるもので最大7台デージチェーンでの接続、また
容量80MB/130MBさらに60×120×243(mm)のコンパクト
IT X130Sをあなたはどのように
SELECT
POWER
itec
IT X130S
itec
IT X80S
IT X80S ¥128,000
国産高精度SCSIドライブ搭載
容量80MB
平均アクセスタイム20ms
接続ケーブル添付
IT X130S ¥158,000
国産高精度SCSIドライブ搭載
大容量130MB
平均アクセスタイム20ms
接続ケーブル添付
イテック株式会社
本社 〒550 大阪市西区新町1丁目25番14号 TEL:(06)532-0216 FAX:(06)532-7253
関西営業所 〒550 大阪市西区新町1丁目25番14号 TEL:(06)532-0120 FAX:(06)532-0543
関東営業所 〒275 千葉県習志野市谷津1丁目12番5号 TEL:(0474)77-7564 FAX:(0474)73-2759
名古屋営業所 〒460 名古屋市中区大須2丁目28番31号 TEL:(052)212-1487 FAX:(052)212-1627
インフォメーションセンター
TEL:(06)532-0320

新・生・活スタートセール 3/21(木)〜4/7(日)迄

★期間中ツクモ全店店頭にて¥100.000以上お買い上げの方に、ツクモイメージガール越智静香ちゃんのテレホンカードor写真を先着1,000名様にプレゼントいたします!!
★ツクモパソコン本店は更に充実して、IBM、PS55Z、NEXT、AXパソコンMIDI、FMノートパソコン等、豊富に取り扱っております。
掲載商品2万円以上送料無料!!(離島を除く)

春のパソコンラックフェアー 3/15(金)〜3/31(日)

見てさわってお確かめ下さい。 パソコン本店3F イベントフロアにて

- 1ドライブタイプ。
- 3.5インチ2DD/2HD対応ドライブ使用。
- ユーティリティソフト付属。

春のわんさかフェアー

新製品が勢揃い!ビジネスソフト・ホビーソフトの新作で盛り沢山。いろいろな催し物を用意しておりますので、是非お立ちよりください。

開催日 4月13日(土)・14日(日)
場所 ツクモパソコン本店2階フロア及び3階イベントフロアの予定。

得 メモリーボード(X68000用)

1MB増設RAMボード(ACE/PROシリーズ用)
ツクモ特価¥19,000 (消費税別途¥570)
2MB増設RAMボード ツクモ特価¥37,000 (消費税別途¥1,110)
4MB増設RAMボード ツクモ特価¥64,000 (消費税別途¥1,920)
※計測技術のメモリーボードも取り扱っております。価格については、お尋ねください。

X68000シリーズ

ツクモSUPERセット

CZ-604C-TN ………¥348,000
CZ-606D-TN ………¥ 79,800
IT X130S(TN) ………¥158,000
合計定価¥585,800
ツクモ特価¥488,000

SUPER CZ-604C

SCSIインターフェイス内蔵タイプ
*オプションの80MB内蔵ハードディスクを追加する事によってSUPER HDに変身!

PRO II CZ-653C 定価¥285,000
CZ-663C 定価¥395,000
EXPERT II CZ-603C 定価¥338,000
SUPER HD CZ-623C 定価¥498,000
ツクモ特価販売中!

ワクワクコンピュータミュージック

Aセット
- CM-32L ………¥69,000
- SX-68M ………¥19,800
- MusicstudioMu-1 Ver1.4 ………¥19,800

合計定価¥108,600
ツクモ特価¥88,000 (消費税別途¥2,640)
クレジット例(18回払・税込) 初回¥7,223+月々¥5,600×17回

Bセット
- CM-64 ………¥129,000
- SX-68M ………¥19,800
- Musicstudio Mu-1 Ver1.4 ………¥19,800

合計定価¥168,600
ツクモ特価¥138,000 (消費税別途¥4,140)
クレジット例(24回払・税込) 初回¥7,603+月々¥6,900×23回

※「Misicstudio PRO68K Ver2.0又は「Music PRO68K」(MIDI)のソフトの場合には、¥9,500プラスになります。また、これらのソフトウェアがバージョンアップにより価格が変更になった場合には変更となります。

ローランド 追加オプション機器
ステレオマイクロモニター CS-10 ………定価¥17,000
MIDIキーボードコントローラー PC-200 定価¥36,000
はなうたくん CP-40 ………定価¥33,000

開発ツール

- C Compiler PRO-68K ………定価¥44,800
- SX-WINDOW ………定価¥6,800
- XBAS TO C CHECKER PRO-68K ………定価¥9,800

ビジネスツール

- Hyper WORD ………定価¥39,800
- Multi WORD NEW ………近日発売予約受付中!
- FIXER Ver4.0…ツクモ特価¥15,800(消費税別途¥474)

アートツール(ハードウェア)

A4サイズ カラーイメージスキャナー…台数限定¥128,000
HGS-68 ファインスキャナーX68 特価¥31,800(消費税別途¥954)
CZ-6VT1 カラーイメージユニット ………定価¥69,800
CZ-6BV1 ビデオボード ………定価¥21,000
CZ-8PC5 48ドットカラー漢字熱転写プリンター NEW 定価¥96,800

アートツール(ソフトウェア)

CANVAS PRO-68K ………定価¥29,800
Z'sSTAFF PRO-68K Ver.2…ツクモ特価¥46,400(消費税別途¥1,392)
マジックパレット ………ツクモ特価¥15,800(消費税別途 ¥474)

ツクモは SONY MO認定店です!

光磁気ディスクユニット

ソニー NWP-539N(縦型) ……¥440,000
NWP-559(横型)
シャープCZ-6BS1(SCSIボード)·¥29,800
SCSIケーブル ………¥10,000
光磁気ディスク…サービス(¥30,000)
合計定価¥509,800
ツクモ特価¥398,000 (消費税別途¥11,940)
クレジット例(48回払・税込) 初回¥13,872+月々¥11,000×47回

ツクモグローバルカード

大/好/評/入/会/者/募/集/中
18才以上なら学生でもOK!!

国内・外で大活躍!
使って便利、持ってて安心!ツクモグローバルカードはジャックス・VISA、セントラル・マスターとの提携カードです。ツクモ各店でのお買い物がらくらくできるうえに、国内はもとより海外でのショッピングもOK!
お問い合せは……
☎03(3251)9898又は各店で!

X68000専用ハードディスク

目玉品! 1台目専用
40MB SASIタイプ ツクモ特価¥59,800(消費税別途¥1,794)

SASIタイプ
アイテック IT X680 (80MB)定価¥198,000
ツクモ特価¥89,800 (消費税別途¥2,694)

IT X680 (カラー:ブラック・グレー)

SCSIタイプ
アイテック IT X80S (80MB)定価¥128,000
ツクモ特価¥99,800(消費税別途¥2,994)
アイテック IT X130S (130MB)定価¥158,000
ツクモ特価¥125,800 (消費税別途¥3,774)
IT X130S

※X68000SUPERシリーズ以外の機種はCZ-6BS1(SCSIボード)定価¥29,800が必要です。

ツクモ通販センター フリーダイヤル受注専門 **0120-377-999** 商品についてのお問い合せは各店店頭又は…☎03(3251)9911へ

ツクモ買取りセンター ☎03(3251)9977 AM11:00〜PM5:00迄

秋葉原各店
営AM10:15〜PM7:00
休毎週木曜日(3/21を除く)
★表示価格には消費税は含まれておりません。

ツクモは「スーパーX PRO SHOP」です。
PRO STAFF ツクモ
九十九電機株
〒101-91 東京都千代田区神田郵便局私書箱135号
★商品のご注文は在庫確認の上お願いします。

ツクモパソコン本店2F ☎03-3253-5599 (担当/荒井)
便利で安心な通信販売
ツクモ通販センター☎03-3251-9911

- ツクモAV/カメラ館B1 ☎03-3254-3999(担当/川名)
- ツクモニューセンター店 ☎03-3251-0987(担当/福地)
- ツクモ5号店 ☎03-3251-0531(担当/森)
- 名古屋1号店 ☎052-263-1655(担当/吉高)
- 名古屋2号店 ☎052-251-3399(担当/横山)
- ツクモ札幌店 ☎011-241-2299(担当/田口)

カード払い	全国代金引き換え配達	クレジット払い	現金書留払い	銀行振込払い	各種リース払い
通信販売での御利用カード、ツクモグローバルカード、VIPカード、セントラル、ジャックス※御本人様より電話で通信販売部へお申し込み下さい。	お申し込みは☎03-3251-9911へお電話1本!配達日の指定もできます。	月々¥3,000以上の均等払いも頭金なし、夏・冬ボーナス2回払いも受付中!	〒101-91 東京都千代田区神田郵便局私書箱135号 ツクモ通販センター Oh!X係	事前に☎でお届け先をご連絡下さい。富士銀行 神田支店(普)No.894047 ツクモデンキ	くわしくは各店にお問い合せ下さい。ケースに合わせてご相談にのらせて頂きます。

いたれりつくせり本(BOOK)フェア

—今のあなたのままで、もっとパソコンがわかる—

◆開催書店一覧◆

地区	書店名	期間		TEL
〈北海道〉				
札幌市	紀伊國屋書店札幌店	4/30〜5/30	A	011-231-2131
〃	旭屋書店札幌店	3/30〜4/30	B	011-241-3007
旭川市	旭川富貴堂本店	3/1〜4/15	B	0166-26-3481
〈東　北〉				
青森市	成田本店	3/20〜4/19	B	0177-23-2431
弘前市	紀伊國屋書店弘前店	3/20〜4/19	A	0172-36-4511
八戸市	金入番町店	3/1〜4/15	B	0178-46-1811
盛岡市	さわや書店	3/20〜4/19	B	0196-53-4411
仙台市	金港堂	3/20〜4/19	B	022-225-6521
〃	金港堂ブックセンター	4/11〜5/10	A	022-223-0979
秋田市	三浦書店	3/1〜3/20	B	0188-33-8131
山形市	八文字屋本店	3/15〜4/15	B	0236-22-2150
酒田市	みずほ八文字屋	3/20〜4/19	B	0234-22-3144
福島市	岩瀬書店コルニエツタヤ店	3/15〜4/15	A	0245-21-2101
いわき市	ヤマニ書房本店	3/20〜4/19	B	0246-23-3481
〈関東・首都圏〉				
水戸市	ツルヤブックセンター	3/20〜5/30	B	0292-25-2711
浦和市	須原屋本店	3/15〜4/30	A	048-822-5321
〃	須原屋コルソ店	3/15〜4/15	B	048-824-5321
船橋市	芳林堂書店津田沼店	3/30〜4/30	B	0474-78-3737
柏市	西口アサノ	4/1〜4/22	B	0471-44-2111
横浜市	有隣堂トーヨー店	4/5〜5/6	A	045-311-6265
〃	栄松堂相鉄ジョイナス店	3/15〜4/30	A	045-321-6831
川崎市	文教堂書店溝ノ口店	3/8〜3/30	B	044-811-8258
厚木市	有隣堂厚木店	3/15〜4/30	A	0462-23-4111
〈東　京〉				
千代田区	三省堂書店神田本店	4/1〜4/30	A	03-3233-3312
〃	書泉グランデ	4/1〜4/30	A	03-3295-0011
〃	ラオックス ザ・コンピューター館	3/15〜4/15	A	03-5256-3111
〃	丸善お茶の水店	3/16〜4/14	A	03-3295-5581
〃	Bit INN東京	3/15〜4/15	B	03-3255-4575
〃	T-ZONE	3/15〜4/15	A	03-3257-2660
中央区	八重洲ブックセンター	3/7〜3/27	A	03-3281-1811
〃	丸善日本橋本店	3/20〜4/20	A	03-3272-7211
渋谷区	大盛堂書店	3/20〜4/30	A	03-3463-0511
〃	旭屋書店渋谷店	3/1〜4/8	B	03-3476-3971
〃	紀伊國屋書店渋谷店	5/1〜5/31	A	03-3463-3241
新宿区	福家書店新宿センタービル店	3/10〜4/13	B	03-3345-1246
豊島区	芳林堂書店池袋本店	3/1〜4/30	A	03-3984-1101
調布市	真光書店	3/1〜3/30	A	0424-87-2222
町田市	久美堂小田急店	3/20〜4/30	A	0427-27-1111
〈甲信越・北陸〉				
甲府市	文教堂書店甲府店	3/20〜4/19	B	0552-22-4600
長野市	平安堂長野店	3/20〜4/30	B	0262-26-4545
〃	長谷川書店	3/15〜4/30	B	0262-26-2122
松本市	ブックスロクサン	3/15〜4/30	B	0263-35-5555
〃	アクロスブックセンター松本店	3/20〜4/30	B	0263-32-5733
新潟市	紀伊國屋書店新潟店	3/15〜4/15	B	025-241-5281
長岡市	覚張書店	3/20〜5/20	B	0258-32-1139
富山市	瀬川書店	3/15〜4/15	B	0764-24-4566
高岡市	文苑堂書店	3/1〜4/30	B	0766-21-0333
金沢市	うつのみや片町本店	3/1〜4/20	A	0762-21-6136
〃	書林香林坊本店	3/1〜3/15	B	0762-20-5011
野々市町	王様の本本店	3/15〜4/30	B	0762-46-5325
福井市	勝木書店	3/1〜4/30	B	0776-24-0428
〈東　海〉				
静岡市	静岡谷島屋呉服町本店	3/15〜4/30	B	0542-54-1301
沼津市	吉野屋	3/8〜5/10	A	0559-23-5676
浜松市	浜松谷島屋連尺店	3/8〜4/8	B	053-453-9121
名古屋市	三洋堂書店上前津店Σ	3/8〜4/10	A	052-251-8334
〃	三省堂書店名古屋店	4/1〜4/20	A	052-562-0077
〃	丸善名古屋支店	4/4〜4/30	A	052-261-2251
〃	丸善ブックメイツセントラルパーク店	3/8〜4/7	B	052-971-1231
〃	星野書店近鉄ビル店	3/18〜4/30	B	052-581-4796
豊橋市	精文館書店	3/15〜4/30	B	0532-54-2345
岐阜市	自由書房	4/10〜5/15	B	0582-65-4301
津市	別所書店11ビル店	3/8〜5/30	A	0592-24-1014
四日市市	文化センター白揚	3/7〜4/7	A	0593-51-0711
〈近　畿〉				
京都市	アバンティブックセンター	3/8〜3/30	A	075-682-5031
奈良市	駸々堂書店奈良大丸店	3/7〜4/30	A	0742-26-6241
大阪市	旭屋書店本店	3/1〜3/30	A	06-313-1191
〃	紀伊國屋書店梅田店	3/26〜4/16	A	06-372-5821
〃	駸々堂Ver99	3/8〜3/30		06-353-4011
神戸市	ジュンク堂センター街店	3/15〜3/30	A	078-392-1001
和歌山市	宮井平安堂	4/1〜5/31	B	0734-31-1331
〈中　国〉				
広島市	紀伊國屋書店広島店	4/1〜4/30	A	082-225-3232
山口市	文栄堂	3/20〜4/30	B	0839-22-5611
松江市	ブックセンター今井学園通店	3/22〜4/30	B	0852-31-8099
鳥取市	富士書店	3/15〜4/15	B	0857-23-7271
〈四　国〉				
徳島市	小山助学館本店	3/20〜4/30	B	0886-54-2135
高松市	宮脇書店本店	3/20〜4/30		0878-51-3733
高知市	金高堂書店	3/10〜5/10	B	0888-22-0161
松山市	紀伊國屋書店松山店	3/20〜4/30	B	0899-32-0005
〈九　州〉				
福岡市	りーぶる天神	3/20〜4/30	A	092-713-1001
〃	紀伊國屋書店福岡店	4/22〜5/12	A	092-721-7755
北九州市	ナガリ書店	3/16〜4/21	B	093-521-1044
別府市	明林堂書店別府本店	3/1〜4/30	B	0977-23-2183
大分市	パルコブックセンター大分店	3/30〜4/30	B	0975-35-0643
宮崎市	寿屋宮崎店	3/25〜5/15	B	0985-27-4111
佐賀市	金華堂北バイパス店	3/15〜4/30	B	0952-32-1965
長崎市	好文堂書店	4/1〜4/30	A	0958-23-7171
佐世保市	BKアカデミー	4/14〜6/30	A	0956-48-5503
熊本市	長崎書店	3/30〜4/30	B	096-353-0555
鹿児島市	BOOKSみすみ	3/30〜4/30	B	0992-57-1011
那覇市	文教図書	4/19〜5/30	B	098-862-1201

第3回アマチュアCGAコンテスト 入選作品発表!!

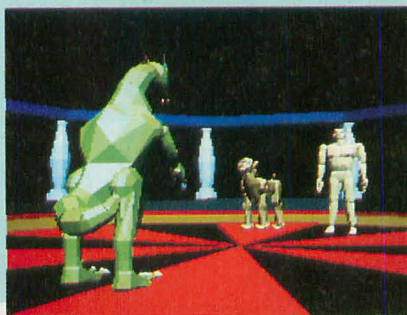
いよいよ皆さんお待ちかね，第3回アマチュアCGAコンテストの審査結果発表です。今年は去年に比べて全体的にレベルが高かったようで，なかなか見ごたえのある作品が揃いました。年々盛り上がりを見せるCGAコンテスト。それでは，さっそく入賞作品を紹介していくことにしましょう。

グランプリ・最優秀映像賞・作品賞
SWORD

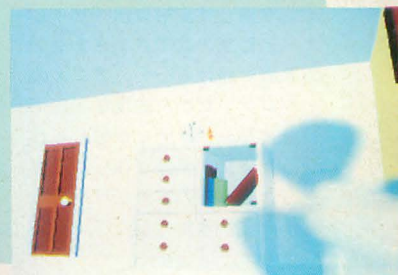
森山 知巳

時間　　　：3分52秒
使用機種　：X68000
使用ソフト：DōGA CGAシステム

ふと目をさますと，まったく見知らぬ場所。目の前の扉を開け，足を踏み入れると，そこには……。作者は，先月号でも紹介したように日本画家。さすがにプロだけあって映像作品として群を抜いたセンスで，ほかの作品を圧倒した。

レベルの高かったと噂される今年のコンテストのなかでも，グランプリとして自信をもってご覧いただける作品である。

最優秀技術賞・作品賞
CLOCK

京大マイコンクラブ：上原 哲太郎

時間　　　：6分50秒
使用機種　：X68000
使用ソフト：DōGA CGAシステム

振子時計に会うために，動き始めたCLOCKの前にさまざまな難関が立ちふさがる……。きわめて完成度が高く，動きが実に表現力豊かである。また，先月号でも紹介したように，CGAの新しい技術が随所に盛り込まれている。

映像賞
Origin
HI-side：伊藤 英基/平田 剛

時間　：7分38秒
使用機種　：X68000
使用ソフト：DōGA CGAシステムほか

“教育におけるCGAの活用”という意図で制作された作品。特に後半の授精のシーンは，NHKの“人体”を思い出させ，とてもパソコンで作られたとは思えないほど見事である。また，前半の解説もナレーションがすばらしい。

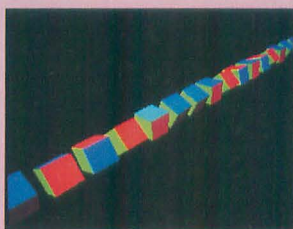
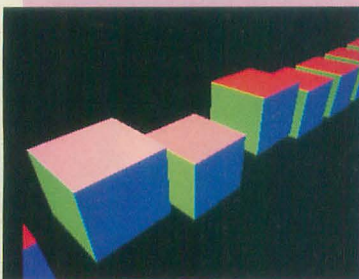
エンターテイメント賞
おはようございますの帽子屋さん
小島 禎樹

時間　：2分20秒
使用機種　：Macintosh＋業務用VTR
使用ソフト：SWIVEL3D

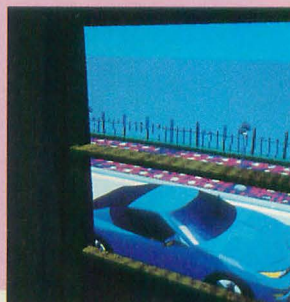
谷山浩子の同名の曲に合わせて映像を作り，NHKの“みんなの歌”風にまとめている。Macintoshを使用しており，個人でコマ撮り可能の業務用VTRを購入された。さすがに編集は正確で，音楽と完全に一致している。

芸術賞
connection'90
京大マイコンクラブ：須田 淳

時間　：1分18秒
使用機種　：X68000
使用ソフト：DōGA CGAシステム

この作品は，直線上のある1点に加えた力が，どのように伝達されていくかをシミュレーションしたものである。現実には見ることができない，回転の伝達などもよくわかる。CGAならではの作品といえる。

特別賞
TORNADO
文月工房：文月 涼

時間　：47秒
使用機種　：X68000ほか
使用ソフト：DōGA CGAシステム
　　　　　　Z's STAFF

自分の理想の車をデザインし，そのCMを制作したものである。TV用に30秒ちょうどに編集されている。車に対して徹底してこだわっており，見えないような細部や，ボディの微妙なラインを出すために，何十枚もの図面を作成した。

エンターテイメント賞
ゲッピーロボ
京大マイコンクラブ：石井 源久

時間　：2分35秒
使用機種　：X68000
使用ソフト：DōGA CGAシステム
　　　　　　マジックパレット

3機合体変形ロボ「ゲッピー」は京の都でさまざまな敵と戦い続ける……。アニメのパロディだが，表現などに十分オリジナリティが感じられる。また，模型が作られており，すべての変形が可能であることが確かめられている。

特別賞
ブランコ
萌企画：島田 弘明

時間　：9分22秒
使用機種　：X68000
使用ソフト：DōGA CGAシステム

宇宙船に乗って旅立った恋人をひたすら待つ。月日は流れ……。娘さんがキャラクターデザイン，奥さんが演技指導など，家族そろってCGAを制作しているのがほほえましい。ストーリー性も高い。

第3回 アマチュアCGAコンテスト 入選作品発表!!

入選
It could happen to you!
琉球大学ニューメディアクラブ：賛 良則

時間　　：２分21秒
使用機種　：X68000，PC-9801
使用ソフト：DōGA CGAシステムほか

「メディ坊」が植えた花はただの花ではなかった。どんどん増殖し……。クラブのマスコットキャラクター「メディ坊」の飛び出したまゆげなど，豊かな表情が楽しい。

入選
デスペラード
京大マイコンクラブ：横山 浩之

時間　　：１分52秒
使用機種　：X68000
使用ソフト：DōGA CGAシステム
マジックパレット

ひとつの星の存亡をめぐるSF大作。同クラブにて現在も書き続けられている小説を映像化したもの。ただしこの作品はあくまで予告編のみであり，本編はない。お絵描きツールによる人物描写が一興。

入選
幸福の皇子
team Eggs：伊藤 圭一

時間　　：２分50秒
使用機種　：X68000
使用ソフト：DōGA CGAシステム
Z's STAFF

皇子は姫を助けるために，怪物の棲む城へ向かうが……。カメラワークや編集も非常に凝っており，TVアニメの手法などをよく研究している。また，パワフルな，手による描き込みも見もの。

入選
幻燈
下田 紀之

時間　　：３分45秒
使用機種　：X68000
使用ソフト：DōGA CGAシステム
Z's STAFF

一瞬，脳裏をかすめる走馬燈のようなイメージを，音楽に合わせて幻想的に演出している。3Dよりも手描きのCGに重点を置いており，CG作品の制作が目的ではなく，作品制作にCGを利用していると割り切っているようだ。

参考入選
祭灯
電気通信大学まに研：西之園 修

時間　　：１分15秒(オリジナルは８分55秒)
使用機種　：X68000，PC-286
使用ソフト：MUSCAT（自作）

色まで塗って完成しているセルや背景をスキャナで取り込んで，CRT上に重ねて表示し，８mmカメラでコマ撮りをしている。いわゆるCGA作品ではないのだが，この手法は新しい表現に応用できる可能性がある。

入選
HEART
寺尾　響子

時間　　：４分19秒
使用機種　：X68000＋トランスピュータ
使用ソフト：DōGA CGAシステム
サイクロン

ハートのエースがひとり静かに笛を吹く……。先月号にも掲載したように，作者の方の職業は主婦兼イラストレーター。作品はほとんど静止画に近いカットで構成されており，それらが，すべて絵として成り立っているのはさすがだといえよう。

入選
小鳥の朝食
木村 良和

時間　　：１分33秒
使用機種　：X68000
使用ソフト：Z's STAFF
DōGA CGAシステム

親鳥は，ヒナたちのエサにト音記号を運んでやるが……。すべての絵を，１枚１枚お絵描きツールで描き，CGAシステムでアニメーションさせている。作者はグラフィックデザイナーであり，色使いにテクニックを感じる。

参考入選
手紙
北川 博之

時間　　：３分25秒
使用機種　：X68000
使用ソフト：DōGA CGAシステム
Z's STAFF

失恋した女の子が，ひとり公園でたたずんでいると……。今回の応募のなかで唯一の，実写とCGの合成作品。もともとオリジナルは13分という大作なのだが，コンテスト用にということで作者自身に再編集していただいた。映像と音声との同期，音質やタイミングの取り方などにも力を入れている。

決定！1990年度 GAME OF THE YEAR

さあ，1990年度GAME OF THE YEARの発表がやってまいりました。今年もまた賑やかだったゲーム界。どの作品も総じて面白さのレベルが高かったようで，読者の皆さんからさまざまな声が届きました。さらにアンケートハガキ投票が加わったこともあって得票数もグーンとアップ。この激戦を勝ち抜き，晴れて受賞の栄誉に輝いたのがこのゲームたち。ここはひとつ惜しみなく誉め讃えてあげようじゃありませんか。それでは受賞作の発表です，どうぞ！

編集協力　浦川博之・山田純二・古村聡・金子俊一

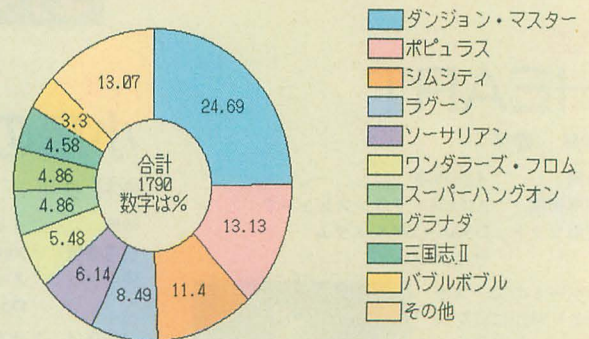
Oh!X ゲーム大賞 ダンジョン・マスター

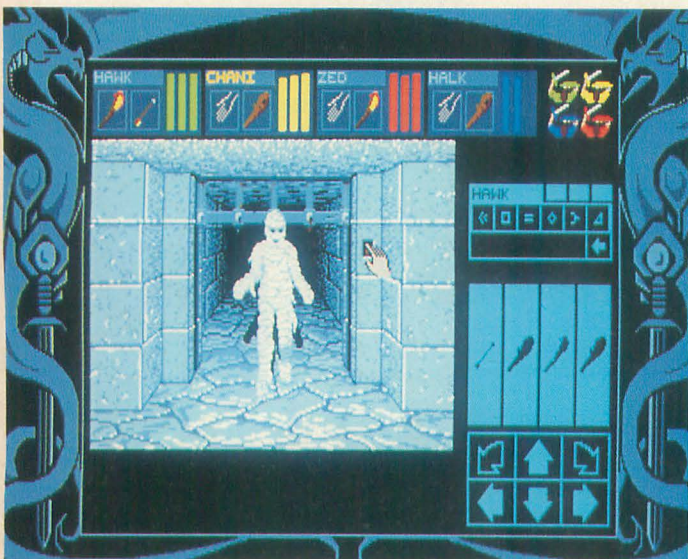
ビクター音楽産業　442票

他の作品を突き放し，ダンジョン・マスターが堂々のOh!X大賞受賞!!　おめでとうございます。

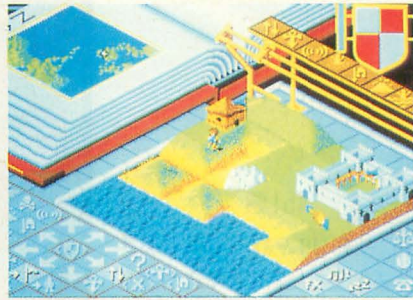
やっぱりといえばやっぱりという気もしますが，1年間まんべんなく話題作が発売になったこの年に，発売時期が早いというハンデをはね返して栄冠を射止めたのだから，たいしたものです。

読者の推薦理由を見てもわかるとおり，これほどまでにダンジョン・マスターの人気が高いのはRPGとしてまったく新しいスタイルを築いたという点にありそうです。落ちているものはなんでも拾って投げられるというアイディアや，徹底したリアルタイム性が，単なる発想ではなくゲームの要素としてがっちりと組み込まれています。マウス操作を失敗したために大切なキャラクターが死んでしまったとか，あわてて逃げて道に迷ったとか，荷物を整理していなかったためにイザというときに大切なアイテムが取り出せなかったとか，そういった体験をしているうちに，プレイヤーがあたかも本物のダンジョンに足を踏み入れているかのような気分になってきます。これはダンジョン・マスターでなければ味わえない魅力でしょう。システムとして高い完成度を持たせたことに対しては脱帽するしかありません。

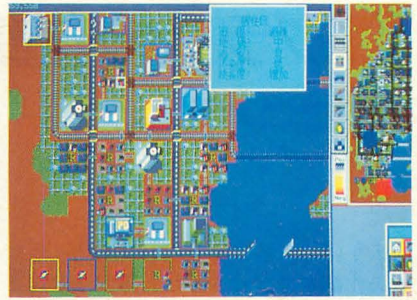
グラフで見る得票数

1990年度Oh!Xゲーム大賞

順位	タイトル	票数	順位	タイトル	票数
1位	ダンジョン・マスター	442票	6位	ワンダラーズ・フロム・イース	98票
2位	シムシティー	235票	7位	スーパーハングオン	88票
3位	ポピュラス	204票	8位	グラナダ	87票
4位	ラグーン	152票	9位	三国志II	82票
5位	ソーサリアン	110票	10位	バブルボブル	59票

2位，3位ともに海外ゲームが入り，去年の海外ゲームの強さをあらためて見せつける結果になりました。これらもダンジョン・マスター同様，今まで日本にないスタイルを武器にしています。

シムシティーは都市を舞台にしたシミュレーション。ただ，都市を眺めながら順調に発展するようにちょこちょこと手を出すだけですが，なかなか思いどおりにならない都市の発展に悪戦苦闘するのがなぜか止められなくなってきます。

一方のポピュラスは，部族同士の対立にスポットをあてててゲーム化しました。相手の住民にさまざまな天変地異を起こしてイジワルできるというのがポイント。最初は「イケナイなあ」と思っていても，相手の容赦ない攻撃にカッとして反撃しているうちに，次第に楽しんでいる自分に気づいてしまうというわけです。

ポピュラス

上位3作いずれもまったく新しい発想のもとに作られているので，Oh!X大賞のこの結果は従来の硬直化したゲームスタイルにユーザーが変化を求めている表れということができるでしょう。

その一方で，ラグーン，ワンダラーズ・フロム・イースなどのアクティブRPGの人気も根強いものがありました。とはいえ決定版を欠き，次世代のアクティブRPGに求められるものを摸索していた1年だったという感があります。

シムシティー

全体的にTOP10の年間チャートと似通った顔ぶれになりました。海外ビッグタイトルとそれを追いかける国産RPG，それにビデオゲームの移植とX68000専用のゲーム……。来年もこの構図はおそらく変わらないでしょうが，勢力バランスがどう変わるかは注目したいところですね。

受賞のお言葉

ビクター音楽産業
ニューメディア開発部　宣伝
小森治信さん

このたびは栄えあるOh!Xゲーム大賞をいただき，本当にありがとうございます。

アメリカやヨーロッパで大ヒットしたこのリアルタイム3DRPGの面白さが，日本のX68000ユーザーにも受け入れられたことは，さすがユーザーのレベルが高いと再確認した次第です。

今後も海外の楽しい興奮するソフトやオリジナルソフトを発売していく予定です。

ぜひまたこのような賞をいただけるように頑張りますので，ご期待ください。

ということで，発売中の「続ダンジョン・マスター　カオスの逆襲」もよろしく。

読者からの推薦理由

◆初めてプレイしたとき衝撃を受けた。これぞパソコンでしか遊べないゲームである。　赤城　豊和(23)兵庫県

◆マウスが壊れるほど，熱くなれるぜ。　三宅　永至(16)岡山県

◆こんなにゾクゾクしたゲームは今までなかった。　梶田　真二(16)奈良県

◆モンスターが突然出てきて，心臓が飛び出そうになったのはこのゲームだけだ。　小笠原　洋(16)東京都

◆手に汗握る緊張感100点，迷路のパズル度100点。　伊藤　盛人(17)福島県

◆RPGをまったく新しい手法で実現した画期的な作品。植木　正幸(22)神奈川県

◆あのリアルな感動がとても良い。　今村　雄治(18)岐阜県

◆日本人としてはつらい！　とはいえいちばん面白かった。　森田　宣幸(19)宮城県

ダンジョン・マスターの推薦理由

空想の世界で遊ぶ。世間では大袈裟な機械を身に着けて暴力的に人を巻き込むバーチャルリアリティと称する機械が流行っているらしい。しかし，問題はソフトである。当たり前である。しかし，このマルチメディアといえば図鑑や百科事典しか作れないような貧困なお国柄では，それが何かを表現するような文化へ辿りつくのは銀色の遥かな道である。山の彼方の空遠くである。そこで暴れていたのは「幸い」ではなく，ダンジョン・マスターであった。バーチャルリアリティといえばスターツアーズであるが，あーゆー大仕掛けをしなくとも，粗いパソコンの画面でもサンプリングの些細な音でも我々は空想世界に遊べるのである。ダンジョン・マスターは4人のキャラクターはどれもプレイヤーではないという点において，実にうまく胡蝶の夢を見させてくれた。ダンジョン・マスターのなかを我々は確かに歩いた。移動したのではなく，歩いたり走ったりしたのである。汗をかき，慌てたのである。そこがほかのソフトと違うところだ。だからダンジョン・マスターは，とっつきにくいとか難しいとかいわれながらも，人々に愛されるのである。アメリカ人は戦争好きだけど，やることはちゃんとやる。僕らが遊びたいのは得体のしれないおもちゃの国ではなく，仮想現実な世界なのだ。(K)

グラフィック賞
ラグーン

ズーム　451票

去年のジェノサイドに引き続き，ズームの作品が選ばれました。勝因はなんといってもX68000専用という点でしょう。256×256という粗いグラフィック画面を使ってはいるものの，色使いのうまさとアニメーションの迫力でうまくカバーしています。キャラクターの対決といい，ボスキャラの激しい動きといい，X68000専用設計の利点を生かし迫力に満ちた画面を見せてくれました。

2位のワンダラーズ・フロム・イースのほうが溶岩や滝などの表現テクニックにかけては上だったようですが，もとがPC-8801用とあってやや小さくまとまった感があるのが惜しまれます。3位のグラナダもX68000専用。2面の雲の隙間から地上が見えるシーンをあげて推薦してくる人が多かったです。

全般的に美しさよりも動きのある画面のほうが評価される結果となりました。

グラフィック賞

1位	ラグーン	451票
2位	ワンダラーズ・フロム・イース	366票
3位	グラナダ	224票

読者からの推薦理由

◆マップの広さを感じさせないくらいキレイだから。　市來　敏(18)大阪府

◆256×256とは思えない。
大石　武(16)京都府

◆ビジュアルシーンがえらく凝っている。さすがズーム！
本田　英雄(21)埼玉県

スタッフの感想

最近のゲームを語るうえで「グラフィック」はきわめて重要なものである。まあ，「グラフィック」はゲームの「顔」ともいえる。また，このゲーム業界は深刻なデザイナー不足に悩まされている。そういった状況のなかで，この「グラフィック賞」の受賞は大変名誉であると思う。今後もこの「顔」を守りつつゲーム内容のほうも一層精進していただきたいものである。　(善)

音楽賞
ラグーン

ズーム　415票

音楽賞は，グラフィック賞と同じくラグーンの手に。曲だけで聴けばもっといい曲はいくらでもありますが，たたみかけるような音楽がストーリーの展開やゲームの性格とうまくマッチしていること，BGMとしていい働きをしている点が評価を得ました。

2位のモトスは，ゲーム自体の注目度と比べると非常によく健闘したといえるでしょう。3種類のMIDI音源に対応し，アレンジバージョンも用意するなど「MIDIミュージックのモトス」というイメージを確立したのが強かったようです。

3位のグラナダもMIDI音源対応。聞き逃せないのは爆発音がなかなかスカッとするものになっている点です。

MIDI音源を採用したものも増えてきましたが，ラグーンの受賞などを見ているかぎりまだまだ内蔵音源も捨てたものではないようですね。

音楽賞

1位	ラグーン	415票
2位	モトス	247票
3位	グラナダ	241票

読者からの推薦理由

◆ゲームに音楽がバッチリマッチしていた。　段　宏太郎(19)兵庫県

◆なんたってネコが踊りますから。
山中　正宣(17)三重県

◆効果音ドカーン！　ノリのいい音楽でやっていて楽しい。
田辺　和也(16)神奈川県

スタッフの感想

サウンドメイキングはゲームに生命を吹き込む作業である。そういった意味では「ラグーン」は母親から出てきたばかりの元気一杯の赤子のようだ。

グラフィック賞とこの音楽賞の2部門をさらった「ラグーン」，まさにゲーム2大重要拠点の征服に成功したわけである。これからもさらなる進軍を続け次回作「ファランクス」では「ベストゲームデザイン賞」の制覇を期待させる。　(善)

プログラミング技術賞
ナイアス

エグザクト　361票

今年のプログラミング技術のキーワードはズバリ「ラスタースクロール」でしょう。X68000にはファンタジーゾーンから登場し，1990年に入ってからはゲームセンターで「ダライアスII」の画面処理が話題になったこともあって一気に画像処理技術のトレンドとなりました。

それをこれでもか，これでもかと見せつけてくれたのがナイアス。異次元空間の表現に使ったり，半透明処理と併せてワープを表現したりとレベルの高さはとてもルーキーの作品とは思えないほど。

2位にはそのラスタースクロールを初めて本格的にゲームに応用したスーパーハングオン，3位には超巨大キャラが飛びかう3Dシューティング，メタルサイトが選ばれています。

また，1990年12月号で紹介した横内君制作のX1turbo用グラディウスも，ノミネート6作に迫る支持を得ていました。

プログラミング技術賞

1位	ナイアス	361票
2位	スーパーハングオン	320票
3位	メタルサイト	263票

読者からの推薦理由

◆ゲーム性はちょっとと思うけれど，全画面ラスタースクロールは凄い。
大島　靖(23)神奈川県

◆どこかほかのゲームに似ているようだけど，なかなかのテクニック。今後の期待を込めて。
進藤　慎一(20)青森県

スタッフの感想

電源を投入後，数秒後に恥ずかしい回転，拡大処理とともに現れる「ＥＸＡＣＴ」のロゴ。技術だ，これぞ技術だ。ゲーム中のステージ間のビジュアルシーンのワイヤーフレームのデモ。これも技術である。「ナイアス」はいたるところに技術の化学調味料が添加されているのである。しかし，化学調味料の取りすぎは体に良くないのだ。結果的に高血圧を招くのである。(善)

ゲームデザイン賞
シムシティー

イマジニア　521票

Oh!X大賞が海外ソフト対決の場とするならば，ここはもうひとつの対決の場。しかも海外ソフトは斬新なゲームデザインを売り物にしているだけに，このゲームデザイン賞を受賞することはひょっとしたらOh!X大賞以上に価値のあることかもしれません。

海外ソフト3本がまともにぶつかりあった結果，受賞の栄誉に輝いたのはシムシティー。Oh!Xゲーム大賞の雪辱を晴らしました。

市長となって街を作るという発想はもちろん，敵というものがいないこと，箱庭を見るような気分で楽しめ，そのくせ都市設計の仕組みに触れることができるというデザインは確かに独創的。Oh!Xゲーム大賞に輝いたダンジョン・マスターも一応RPGであり，シムシティーに比べるとインパクトの点で及ばなかったようです。

ゲームデザイン賞

1位	シムシティー	521票
2位	ポピュラス	471票
3位	ダンジョン・マスター	342票

読者からの推薦理由

◆このテのゲームは，またべつの面白さがあるから。
大道　顕二郎(16)岩手県

◆創造と破壊のどちらも味わえる。
井戸　直樹(20)岐阜県

◆市長の立場で遊べるのがいい。
宮島　雅史(15)長野県

スタッフの感想

シムシティーには勝敗がない。どんな街でも，街としてのなりを持つ。全滅もないからゲームとしては刺激が少ないかもしれない。が，シムシティーのデザインが優れているのは「挑戦→障害→解決」の流れが，プレイヤーのイメージしだいで実に多様な展開を生みだすということだ。大都市になれば，それだけ複雑な局面と真剣勝負しなくてはならない。ガンバレ猪木！　(晋)

主演キャラクター賞
ポピュラスのナイト

イマジニア　43票

このナイトが家を焼いて周囲を焦土にしていく姿に屈折した笑いを浮かべた人も多いはず。ポピュラスの中の主役は誰だと聞かれたらやっぱりこのナイトになってしまうというわけです。悪の魅力が勝った1990年の主演キャラクター賞でした。

推薦理由

◆こんなやつが世の中に1人でもいたら大変だろうなあ。大久保 益幸(19)滋賀県

◆我々の内に秘める残虐さを浮き彫りにしてくれたから。小川　保彦(18)東京都

主演キャラクター賞

順位	キャラクター	票数
1位	ナイト(ポピュラス)	43票
2位	暁子さん(SX-WINDOW)	20票
3位	マウス	19票

助演キャラクター賞
ズームのネコ

ズーム　98票

得票数でもわかるとおり，このネコの人気は圧倒的。ゲーム本編に登場するわけではないけれど，ちょこちょこと顔を出すたびになんか気になってしまうという，味のあるキャラクターなのです。今度はどこに登場するか楽しみな人も多いのでは？

推薦理由

謎の多いキャラクター。好物は？　家族構成は？　年齢は？　これらの謎はきっと次回作以降のエンベロープで明らかになろう。

(善)

助演キャラクター賞

順位	キャラクター	票数
1位	ネコ (ラグーン)	98票
2位	怪獣(シムシティー)	29票
3位	マウス	18票

集計を終えて

昨年の海外ビッグタイトルの相次ぐ発売は，日本のゲームシーンに新風を吹き込むことになりました。それは海外ゲーム3作の突出ぶりを見てもわかるとおり，ゲームデザインの大切さ，発想の大切さといったことです。

一方の国産ゲームは，特にアクティブRPGとシューティングゲームにおいて過去のヒット作の影響から抜けきれないままゲームデザインの点では足踏み状態といったところでした。グラフィック賞，音楽賞の2賞を受賞するほどの人気ゲーム，ラグーンでさえ西川善司氏によって「イースの別シナリオのようだ」と評されることからもそれはわかります。ラグーンのAVパワー，ナイアスのラスタースクロールなど，技術的な面では大きく躍進を遂げましたが，それらの技術を使って何を作るのかという問題にぶつかっています。

そこに登場した海外ソフトは，まさにゲーム界の黒船来航といえるでしょう。日本のソフトハウスが今後どうするのか，ゲームデザインに力を注ぐようになるのか，それともダンジョン・マスターもどきやシムシティーもどきを作り始めるのか，ユーザーが注意深く見守るときがきています。

底抜け脱線ゲーム賞　該当作品なし

今年はこの賞への応募が少なかったです。面白いゲームが多かったということですね。それはそれでめでたいな。それでも来ちゃったハガキは勝手にGAME OF THE YEARのほうにちょこっと顔を出してますので，ご覧ください。

◆祝！　ダンジョン・マスター

スキーでもさ，苦労なく滑れる自分の実力にあった斜面より，ちょっとビビるかな，ってくらいの斜面のほうが面白いじゃん。そういうものじゃん。本だって，少し自分には難しいかな，ってくらいのほうが面白いじゃん。これはなんでもそうだよね。

今年は当初に望んだ「海外の面白いゲームをばんばんやりたい！　海外には面白いゲームがたくさんあるんだ！」っていう主張が全面的に受け入れられてしまった。祝！　ダンジョン・マスター！　でも，期待どおりってのはつまらない。誰でも「日本のソフトハウスは何やってんだ」ってのは書くと思うけどそれは間違い。だって，ソフトハウスの問題ではなく，日本人や日本社会のメンタリティの問題だから。

私はまだまだもっと欧米の大人のゲームやばかばかしいゲームを輸入してもらいたいと思う。IBM-PCのゲームをしたいがためにテラが売れてしまうくらいまでね。日本にもA列車IIIとかはあるけど，総じてぶっとび方がまったく足りない。作る側も遊ぶ側も。せっかくパソコンというどうとでもなる仮想世界を手にしたのだから，ぶっとぶことを怖がってはいけない。いけいけGo Go!　の精神だ。いってしまったもん勝ちだ。もっとも勝つのは，いってしまって戻ってきた者だ。これが真実だ。

(荻窪圭)

◆もっと感性を育てようよ

ダンジョン・マスター，これってマウスちょんちょんでリアルタイムに物を取れて投げられて敵と戦えて……。これが日本のRPGだったら，「道具　石　使う」とか「たけひこ　戦う　スライム」とかで3回もウィンドウ開かなきゃなんない。ふう。ものぐさなアメリカ人は一生懸命どうすればラクできるか考えるから。お国柄「Take It Easy」てなもんだし。こういうダイナミックな考え方ってのは，日々仕事にこづき回されている日本人にゃ難しいよね。

シムシティーやポピュラスだって，街を作っちまおうとか，わらわは神じゃ，だもんね。ホント，スケール大きいよ。そこにあるのは感覚的な面白さを追求した結果。グラフィックがどうのとかの，理屈じゃなくて感性に訴えかけるものがあるから。

ゲームで人を楽しませるなら，まず自分が楽しい生活しなくちゃ。もっといろんなものを見て，感性を磨いて，余裕を持たなくちゃ。んなこといっても，きっと働き者の多い我が日本国では無理なのかもしれないけどね。ああ，悲しい性……。(出口香)

またまた勝手に GAME OF THE YEAR

さて，読者の声を一挙に紹介する「勝手にGAME OF THE YEAR」のお時間がやってまいりました。受賞作の推薦理由，受賞できなかった作品の声，自由応募部門賞に加えてゲーム回顧録とスタッフの声で今年は一層にぎやか。コンパの2次会のノリでお楽しみください。

ゲーム大賞　シムシティー＆ポピュラス

▶小社商品だから。

佐川俊治　イマジニア㈱広報宣伝課

音楽賞　ダンジョン・マスター

▶あの無音の感覚，突然の叫び声など音という意味ではとてもインパクトがありました。BGMは鳴りっぱなしにすればいいってものじゃない。　米田　孝(20)北海道

主演キャラクター賞　マウス

▶初代のマウスでアフターバーナー，ダンジョン・マスター，ポピュラス，シムシティーを耐えられればコストパフォーマンスは高いのではなかろうか？

吉村　昇(19)大阪府

主演キャラクター賞　ポピュラスのナイト

▶ナイトには人間の破壊願望をくすぐるモノがある。ニヤつきながらナイトの行動を見ていた人も多いはずだ。

水本　昌宏(18)香川県

主演キャラクター賞　優子（ヴァリス）

▶誰もが5面の手前でセーブしてると思う。

稲富　顕二(19)岡山県

底抜け脱線ゲーム賞

・ラグーン

▶どうして「ジェノサイド」のノミネートがひとつもないのであろうか？　古いのは確かだが，今なお最強だと思うのだが。事実遊びにくる友人たちは，やっぱりこれが一番やな，と口を揃えていう。さすがにゲーム大賞とゲームデザイン賞はダンジョン・マスター以外にありえないが，しかしあの凄まじいまでのBGM，体どころか魂さえもゆさぶるようなBGMは，文句なしに音楽賞に今年も1票！　である。ちなみに底抜け脱線ゲーム賞はいたって普通のゲームであったラグーンに決まりである。これもほかのソフトハウスであったら許せたのだが，あのズームだけに。まぁファランクスはジェノサイド以上にエキサイトさせてくれることを期待しましょう。

陣山　達夫(20)大阪府

・ウルフの出すゲーム

▶清水健太郎　神奈川県

▶丸藤俊之　神奈川県

▶ウルフ・チームの出すゲームは，グラナダやアクシスをプレイしてみてわかるようにEASYモードでも難しいですね。皆さんはサギゲー，ハマルなどといっているが，僕はこういうゲームのほうが好きだー！

大竹　裕(16)千葉県

・ダンジョン・マスターのエンディング

▶苦労をして解いたのに最後の少しの絵には悲しかった。PC-6XX1xxを思い出してしまった。　眞鍋　智博(17)香川県

・A列車で行こう！

▶いや，単にIIIでは脱線しなくなったけど，最初はよく脱線してたな～って。

末吉　克行(22)兵庫県

・F.S.S.ティグナスの冒険

▶3時間でクリアできる手軽さ。タケルの前で待っているときがいちばん楽しかったこと。　山崎　康則(16)北海道

・闇の血族

▶本格推理モノだと思ったのに……。

根内　賢一(20)千葉県

大賞　Yet

▶1990年度のゲームソフトNo.1はYetだと思う。このOh!Xを読んでいる人の半分は，このゲームをプレイしたことがあるはずである。それにあのコストパフォーマンス!!　たった780円であれだけ楽しめたんだから，Oh!Xゲーム大賞はYetにあげるべきである。　奥山　浩司(17)三重県

大賞　遊撃王IIエアーコンバット

私は10人で7ならべをやったときは5戦4勝，モノポリーをやれば悪徳商人と呼ばれるほど性格がよい。よって，今までにも性に合わないゲームは存在していた。代表的なのはウルティマシリーズ。私の性格はアバタールとはほど遠い。シムシティーも比較的その筋のゲームであると思う。市民のワガママに付き合うなんて，とてもじゃないが考えられない。まあ，ハーレムを作っていいというなら話は別だが。

ところが，そんな私でも遊べるシナリオがあったのだ。「1972年スイス・交通渋滞をなくせ」である。特に遊ぶつもりはなかったのだが，なにげなく怪獣を出したり，メルトダウンしたり，ともかく街をめちゃくちゃに破壊したのだ。そうしたら，さすがに街は見る影もないが，交通渋滞がなくなっていたんで褒められてしまった。そりゃ，交通局に金をあげなかったし，火事はほっといたから道はボロボロ。あれじゃオフロード用の4WDくらいしか走れないだろうな。めでたし，めでたし？　(S.K.)

▶なんであの超ウルトラスーパーナイスなゲーム，遊撃王IIエアーコンバットがGAME OF THE YEARにノミネートされていないのか。僕はがっかりした。サイバースティックを使うとなかなか楽しめたのに。やっぱり遅いのが問題なのかな……。

入本　泰光(19)東京都

大賞　ワールドコート

▶ワールドコートはテニスの経験者から見れば恐ろしいほど完成度が高い（私はテニス歴５年)。「友達と一緒にプレイする」という条件のもとでは文句なしでの大賞であると断言する（ポピュラスは時間がかかるし，けんかになるしでだめ)。

松田　英弘(20)京都府

(1)このソフトの定価はいくらで賞　ダイナマイト・デューク

▶あるパソコンショップで新品が3,800円で売っていた。その店の中古コーナーでは4,800円だった。数日後，新品が6,800円になっていた。これっていったい？

(2)どれだけ定価を上げたら気が済むんで賞　全ソフトハウス

▶あんなに大きい自社ビルをもっているのに，オーガ○タは８万本以上も売れたのに，△△と××は斬からかなり流用できるはずなのに，ガイナ○クスのソフトは絵以外は楽に作れるはずなのに，FM-Rに移植する余力があるのに，売り上げ本数は確実に増えているのに，どうして？

福永　浩司(19)大阪府

まだ解けないよー賞　殺意の接吻

▶１月号で殺意の接吻が解けないと投書した私。友人やパソ通仲間からも見たとはいわれたのですが，誰もやはり殺意の接吻については教えてくれず編集部の人にも見離され，今は子供と２人パックマニアっています。おまけに，またひとつ年を重ねてしまった。ぐすっ！　吉田　早苗(29)岐阜県

▲江副　滋　東京都

その他自由応募賞　ブルトン・レイ

▶あのデジタル８色は懐かしすぎて怖い。

白根　茂稔(27)神奈川県

よっこら賞　ダンジョンマスター

▶ダンジョンに入るのが非常に怖いので，ゲームを始めるためには気力をふるいおこし，重い腰をあげねばならない。

宮沢　清太(15)東京都

熱血根性で賞　ハム・スライス（ワールドコート）

▶球に飛び込むのがかわいい。

中野　信(18)三重県

しょせん男はＨで賞　ランス2

▶……おおっ！　いつのまにか手に持っていたぞ！　長谷川　敬之(18)福岡県

老兵は去らねばならないで賞　祝一平

▶やはり，若手の代表（？）である西川さんが祝さんに勝ったから！

▲猪原　裕太　東京都

高清水　弾(19)北海道

自由主義の壁は厚いで賞　天安門スペシャル（PDS）

▶中国へは輸出禁止というこのゲーム。単純な内容ながら笑えました。しかし，社会主義を貫くことはできませんでした。10億人は多い……。　升井　晋也(22)福岡県

ずっと続くで賞　夢幻戦士ヴァリスII

▶いつまでもビジュアルのゲームでしょうね。　清水　義弘(22)沖縄県

めだったで賞　M.N.M Software

▶アルガーナに始まり，1990年のOh!Xの台風の目であった。　松田　徹(15)奈良県

どこいっちゃったんで賞　チェイスHQ

▶期待していたのに。

桜井　良太郎(19)東京都

いつになるんで賞　ネメシス'90

▶'90年５月からずっと8,800円用意してるんですけど。　今井　智雄(17)千葉県

気になっちゃうプログラマ賞　K.Hさん

▶MZ-700用メトロクロスの存在を私は知っている。　上野　壮也(19)大阪府

顔が見たかったで賞　片桐まなみ

▶え？　知らないの？　ナイアスの自機「ASP」のパイロットなんだけど。でも，Human68kのVer.1で，DISKの中身を見れば，彼女のおしゃべりが見られるよ。

伊藤　博明(18)埼玉県

功労賞　横内威信さん

▶X1turboでグラディウスをあそこまで作るのは凄い。黒武者　健一(21)神奈川県

大賞はやはりポピュラス。戦略の多彩さ，発見的な喜び，なにより対戦モード。ちなみに，プロミストランドはつまんない。あらゆる要素が微妙にバランスをとって奇跡的な面白さを醸しているように思われる。「ゲーム」としての完成度は圧倒的。手を加える余地はない。シムシティーはゲームを超えている。逆にいえばゲームではない。ダンマスも凄い。RPGという分野であれだけの衝撃が味わえるとは思ってもいなかった。構築した世界も凄い。大脳皮質にとって「謎」というのは麻薬的な魅力を放つ存在だ。謎。ダンマスは凄い。

グラフィック賞はグラナダ。デモはいらないし１面目が長すぎる。それ以外ではグラナダは非常に美しいゲームだった。

音楽賞はガンマプラネット。ゲームは苦手だが曲はいい（ノミネート時には強く推したのだが)。闇の血族もストーリーとの親和性ではゲーム史に残るデキといっていい。

技術賞はナイアス。「新規参入にしては」とかの条件なしに優秀。単にラスタースクロールだけではない。ゲーム自体もかなり練り込まれている。「卑怯」というのは的外れ。難点は色づかいが悪い，敵が見にくい（そういう特性の敵は除いても)。あれだけ次々と驚かせてくれたゲームも珍しい。見せ場を押さえたデモプレイヤーもうまい（くそー，こういう奴がいるから……)。

ゲームデザイン賞はポピュラスかダンジョンマスターか……と迷うところ。シムシティーが入ってないが，Ａ列車を知っているので斬新さには欠けたため。でも，あえて国産のファーストクイーンを推す。ベストエンディング賞というのがあればそれもあげたい（それにしても，外人はエンディングで感動しないんだろうか？)。

主演キャラクターはワールドコートのハム・スライス。はっきりいってこの娘はよくコケる。コンドル相手ではサーブもスマッシュも比べものにならない。足は遅い。それでも技の限りを尽くして勝利をつかむ。「勝つ！」ではなく「負けない！」というプレイがいい（自分でやってんだけど)。やっぱり手のかかる娘ほど可愛い。　(S.N.)

ゲーム回顧録

その1：カオスが終わってしまった。難しい難しいとフカしていたわりにはあっけなかった（魔法の地図があったからこそという話もある）。とゆーわけで，次の狙いはドラッケンである。まだ何も情報がないのでイマイチ不安だが，期待大である。F29とかF15ストライクイーグルIIもいいが，予定のないものはしょーがない。と洋物ソフトばかりを並べたが，日本のソフトが嫌いなワケじゃない。けどやっぱり技術の差というか気合いの入れ方が違うんだな。
大森　幹雄(18)神奈川県

その2：天下統一が大好きです。大賞には天下統一を書きたかったけれど，ちょいと役不足だからポピュラスにしときました。

天下統一は，光栄と比べればビジュアル面とかではハデなところがないし，文字と数字だけでやるようなところは，戦国マニアの人しか楽しめないから評判はあまりよくないけど，とてもやりがいがあって面白かった。天下統一にはバージョンアップをぜひしてほしい。戦闘シーンに力を入れてもらってグラフィックを増やしてほしいな。でも，そのために処理が重くなると困るしディスクはやっぱり2枚におさえてほしいし，難しいだろうとは思うけど……。もっと斬新な戦国ゲームがどんどん出ればいいと思う。
池本　周央(18)京都府

その3：私は最近タケルで初めてゲームを買いました。その名はビンビン麻雀ピーチエンジェルといいまして，麻雀したさと安さにひかれて買いました。最初は簡単でした。しかし進むにつれてツモのみであがったり，食いタンのみ，トイトイなどでがんがんあがられ，私は四暗刻を狙う……。こんな膠着状態が続き，ある日懲りずにゲームを始めたら，いきなりディスクがクラッシュしてしまった。磁気面に1筋の線がついていた。私の涙のよーに。せめて最後までいきたかった。役満が出したかった。無念！
小栗　一也(18)愛知県

その4：ああ，オレにとってOh!XのGAME OF THE YEARもこれで3回目を迎えるのか。うーん，確かにX68000のソフトはここ1年でぐっと成長したが，去年のアフターバーナーほどの盛り上がりは感じられないのは残念だ。来年はズーム，SPS，電波新聞社を筆頭にX68000を大いに盛り上げてほしい。
高橋　洋(18)静岡県

その5「ゲームと私」：1989年が私とX68000との出会いの年であるならば，1990年はパソコンゲームとの出会いの年であった。

1990年の夏までは私の持っているソフトといえば“DataPRO-68K”だけであったし，「たとえ世間が，X68000はゲーム機であるといおうと私はそういう目的には使わない。ゲームをやりたくなったら自分で開発する」といっていたものであった。

そういえば，春まではデータベースソフトも自分で（しかもBASICで）作ろうとしていたっけ。1990年は挫折の年だったのね。

すべては友人の家で起きた。PC-286で信長の野望をやってしまった私はパソコンゲームの世界にひきずりこまれてしまった。

いつの間にか8月18日に神田でOh!Xを買うと，足が自然と秋葉原へと向いていた。銀河英雄伝説とめぞん一刻完結編（私の趣味の世界だ！）を買って家に帰り，その日は1日中ディスプレイに向かいっぱなしであった。

いつしか私の部屋にゲームソフトがあふれた。ナイトアームズ，夢幻戦士ヴァリスII，第4のユニットシリーズ。私のX68000にゲームではなく道具としての未来はあるのだろうか。
小林　裕昭(21)東京都

その6：今年のゲーム界はアフターバーナーのような衝撃はなかったものの，完成度の高い作品は多くありました。ダンジョン・マスターは日本のRPGとはまったく違うものに感じました。また，グラフィックと共に音楽のほうにも力が入った作品が多く，ラグーンなどは，まさにその代表作といえるでしょう。ほかにも書き切れないほどの良い作品がかなりあり，充実した1年だったといえそうです。
山本　昭治(22)神奈川県

その7：今年私がやったゲームは，ほとんどがアクション，シューティングである。グラフィック，音楽は以前と比べて格段によくなった。どのソフトにも良いところ，悪いところはそれぞれあるが，どのソフトにも共通した不満がある。それは“ポーズ”の機能である。ジョイスティックでプレイしていてポーズをかけようとすると，手と目は必然的にキーボードに向かう。その一瞬のスキに自機があえなくやられていたりすることが多い。まあ，これは2つボタンのジョイスティックではしょうがない。でも，ソル・フィースのような1ボタンしか使わないようなシューティングであれば，ぜひジョイスティックでポーズできるようにしてほしい。ポーズを解除するときはさらに問題だ。解除したとたんにゲームがスタートして，やはり手が戻る一瞬のスキにやられる。どうせなら解除してから3秒くらいしてからスタートしてくれればいいのに……と思う。その間は画面上で“3，2，……”とでも表示してくれればもっといい。あと，私はF1のゲームがやりたい。ウイニングランとはいわないから，せめてF1-Dreamくらいのゲームはやりたい。誰か作らないかなぁ……。もちろん自分でもTRYしてみます。
新藤　慎一(20)青森県

最後に

というわけで，皆さんの声を紹介してきたGAME OF THE YEARもこれでおしまいです。いかがでしたでしょうか？

今年はアンケートハガキで投票を大々的に募集するスタイルを取ってみました。「もっと思ったことを書くスペースが欲しい」という声もあり反省してますが，ともあれ非常にたくさんを反映したGAME OF THE YEARとすることができました。投票してくれた皆さん，本当にどうもありがとうございました。

まあ，「集計を終えて」のところでは，海外ソフトの来襲がなんちゃらかんちゃらといって深刻がっていますが，ゲームの立場なんかをいろいろ考えてしまうスタッフよりも，「いいものはいい！」と素直にいえる皆さんの声のほうがゲームを正しく評価しているのは確か。だからこそGAME OF THE YEARの重みというものがあるのです。とにかく自分の出会った面白いゲームを晴れて表彰してあげることができればそれでいいんですよね。

さて，結果は結果として，これからまた来年のGAME OF THE YEARに向けてゲーマーとゲームの対決が始まります。どんなゲームが登場してくれるのか楽しみに待つとしましょう。

そしてソフトハウスの皆さん，僕らをどひゃーと言わせるゲームを作るべく，がんばってくださいね！　期待してるからね!!　ね!!

プレゼント当選者発表

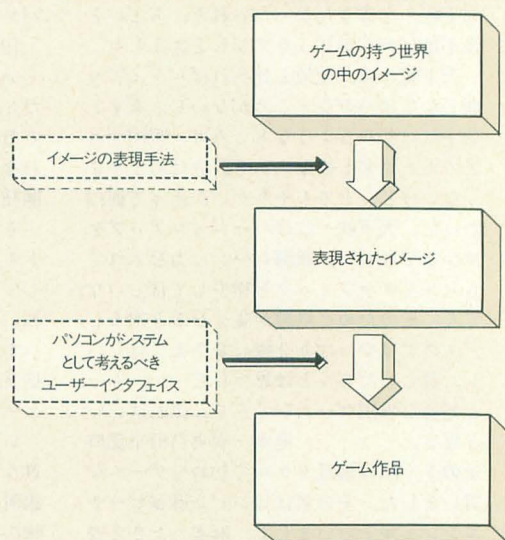
さて，お待ちかねのプレゼント当選者の発表です。まずゲーム回顧録の優秀者に贈るカセットボーイは神奈川県の小林裕昭さんに決定！　自分のポリシーに反してゲームにのめり込んでいくさまが，よく伝わってきました。ハンディスピーカーは大阪府の福永浩司さん，千葉県の根内賢一さんに，カードゲームは北海道の米田孝さん，千葉県の大竹裕さん，東京都の入本泰光さん，岐阜県の吉田早苗さん，京都府の松田英弘さん，大阪府の上野壮也さん，ほか3名の方です。おめでとうございます。

GAME INTERFACE

●特集
人とゲームのインタフェイス

X68000

よりよいソフトウェアの作成に必要なことを知るために，記事としてできることはないだろうか。もちろんソフトウェアの価値は知識や技術だけで語られるものではない。とりわけゲームソフトにおいて重要なのは楽しさや新鮮さなどの感覚的な要素かもしれない。しかし，だからといって作り手のセンスとか感性などという話を持ち出してみても事態は何も解決しない。それらはたいへん重要な問題だが，取り上げるテーマとしてはもう少し普遍的な問題に絞りたい。そこで，今回は「ユーザーインタフェイス」というものに焦点を当ててアプローチをかけてみることにする。それはコンピュータでゲームを行うということを根本から解析するための試みでもある。

ゲームにおけるユーザーインタフェイス

ゲームの持つ世界の中のイメージ
イメージの表現手法
表現されたイメージ
パソコンがシステムとして考えるべきユーザーインタフェイス
ゲーム作品

\CONTENTS

ゲームとユーザーインタフェイス

インタフェイスというのは，なにかとなにかを結びつける接点のことを意味します。パソコンのように多くの技術が結集されたものでは，それら個々の技術をつなぐための技術というのが必要になります。それがインタフェイスの考え方です。

たとえば，楽器をつなぐためのMIDI，ハードディスクや光磁気ディスクをつなぐSCSIなどといった規格がありますが，これらの略語の最後のＩはいずれもインタフェイス(Interface) のＩですね。皆さんの目に触れるものだけでも，コンピュータ本体とディスプレイ，キーボード，ジョイスティック，マウスなど多くのものがつながっていますが，これらをつなぐ回路はすべてインタフェイスと呼ばれています。

ユーザーインタフェイスとは

一方，ユーザーインタフェイスというのは，プリンタインタフェイスとかハードディスクインタフェイスとかいった他の多くのインタフェイスと違って遙かに厄介な代ものです。なぜかというと，ほかのものが機械と機械のインタフェイスであるのに対しユーザーインタフェイスはまさしく人間と機械のインタフェイスであるからなのです。

キーボードを使ってコンピュータに文字を入力したり，画面に表示された文字を目で確認したりすることは，人間と機械との情報伝達，つまりコミュニケーションといえるでしょう。このコミュニケーションの在り方を考えるのがユーザーインタフェイスというわけです。

しかし，キーボードを使ったコミュニケーションでは正確なタイピングや情報の入出力に関する詳しい知識を必要とし，誰にでも簡単にパソコンと対話するというわけにはいきません。そこで，パソコンとの情報のやりとりのためにわかりやすいメニュー表示を用意したり，指でものを指すように目的の場所を指示できる便利な入力装置(つまりマウスのことです)が開発されたりしてきました。

パソコンのなかでこの分野の先駆者となったのはAppleのMacintoshで，ウィンドウシステムというマウスやビジュアル表現を中心とするオペレーションが「ユーザーインタフェイス」という言葉を一気にパソコンのキーワードにしてしまったのでした。最近ではこうしたウィンドウシステムを採用するパソコンが増えてきました。X68000のSX-WINDOWやマイクロソフトのWindows 3.0なども その一種です。

ゲームにおけるインタフェイス

今回のテーマはゲームのインタフェイスなのですが，これもよくいわれるユーザーインタフェイスとさほどの違いはありません。ただ，パソコンのユーザーインタフェイスが主として機械と人間のインタフェイスを指すのに対し，ここでは，ゲームの持つ世界とプレイヤーの情報伝達を取り持つものとしていっそう抽象的な意味を含んでいるといえます。

では，具体的にどういったものがインタフェイスとして問題となるのでしょう。ちょっと思いつくものを挙げてみます。

インタフェイスというからにはなんといっても操作方法に関係するものを思い浮かべる人が多いのではないでしょうか。キーボード，マウス，ジョイスティック，トラックボールなどの入力装置を使った操作法全般ですね。

それから重要なのは画面表示に関するもの。人間が視覚を通じて得る情報量は多く，パソコン自体もディスプレイをユーザーインタフェイスの要として位置づけています。画面の持つ意味は大きく，全体のレイアウトから細部のメッセージ，あるいはウィンドウなどの表示方法，メニューやアイコンなどさまざまな要素があります。また，画面モード，解像度なども含まれるといってよいでしょう。

さて，いま挙げたようなものはなにもゲームに限ったものではなく一般的なユーザーインタフェイスに関するものばかりです。しかし一方で，ゲームの持つ世界そのものをいかにしてプレイヤーに伝えるかという意味でインタフェイスとなるものもあるのです。このような見方ではグラフィックによる画像表現自体もインタフェイスとしての役割を持つと考えられます。これはさらに，表現方法と，それによる表現がゲームの世界を効果的に反映しているかということが問題となるでしょう。

たとえば，RPGのグラフィックを考えてみます。舞台はトップビューか，サイドビューか，はたまた3Dダンジョンか，画像はペイントされたものか，チップで構成されたものか，背景はスクロールか画面切り替えか，などといったさまざまな表現手段が考えられます。そしてそれらの手法によって表現されたキャラクターや背景がゲームの世界にリアリティを持たせることができるかということがいちばんの問題ではないでしょうか。

どこに視点をおくか

インタフェイスというものは特定のものを指す言葉ではありません。コミュニケーションのレベルによってさまざまなインタフェイスがあるのです。人間同士が話をする場合，会話を成立させるという意味では当然語学力がインタフェイスになりますし，深い意志の疎通がなされるためには両者が持つ知識や教養がインタフェイスとなることもあります。この場合の語学力がシステムとして考えるべきインタフェイスであり，知識や教養がゲームのためのインタフェイスとなるわけです。

いずれにしてもユーザーインタフェイスの役割の基本は，ゲームとプレイヤーとのコミュニケーション，すなわち情報伝達にあります。そこで，この問題を扱うには常に２つの方向からの情報の流れに注意してみていかなくてはなりません。つまり，ゲームからプレイヤーへの情報伝達とプレイヤーからゲームへの情報伝達です。もちろん，これらは独立したものとして考えてはいけません。なぜなら，２つの情報の流れによる相互作用でゲームというものが成り立つからです。

一方から情報が伝わると，リアクションが他方に伝わります。これがゲームが進行している状態です。人間はパソコンからの情報に敏感に反応しますが，人間からの情報にパソコンがうまく反応しなければ，ゲームはちぐはぐな状態に陥るでしょう。そうならないためにもソフトウェア側は，難しい要求に応えなくてはなりません。

そしてこれらの問題を考えていくと，結局は「わかりやすさとは何か」ということにつながっていくのです。

よりよいものを作るために

戦うインタフェイス

Nakano Shuichi 中野 修一

ゲームでハマるのとゲームにハマるのでは大きく違います。ゲームにのめり込めたいとは誰もが思うこと。そして，できるなら気持ちよくハマりたい。ではどうすれば？　少し難しい問題だから少しは考えてみる，必要なのはそれだけです。

昔，若手のスタッフから聞いた話。とあるゲームのレビューで画面まわりや操作性について書いたものがあった（いまもよくあるが)。彼はそれを読んだ友達に，シナリオやゲームバランスより見た目や操作性を重視するのかと白い目で見られた，のだそうだ。なるほど。

4つのパターンを考えよう。

1)　中身がよくて見た目や操作性がよい
2)　中身はつまらないが見た目や操作性がよい
3)　中身はよいが見た目や操作性が悪い
4)　中身もつまらなくて見た目も操作性も悪い

1)，4)は明快。2)にも思い当たるものは多いだろう。しかし，3)のパターンは案外少ない。操作性が悪ければ，面白さが伝わるまで触ってさえもらえず，いきなり4)とみなされる場合が多いのだ。

十二間間口の店を広げていて中身がないというのも困りものだが，ユーザーとの接点を大事にしないようなソフトハウスに対して中身の期待ができるだろうか？

ソフトウェアの潮流はGUIを持つ方向に向かいつつある。これは画面が綺麗なほうがいい，ということではない。これまでの機能や性能は落とさず“LOOK and FEEL”のむしろ“FEEL”の部分で快適であることを目指そうとしている（と信じたい)。

だから綺麗に飾れとはいわない。プレイする人のことをもっと考えてほしいだけの話なのだ。

綺麗だと気持ちいい。操作性がいいと気持ちいい。これはわかりやすい。ゲームをやっていれば，操作性の善し悪しは一目瞭然。どれがよい悪いというまでもないし，皆さんのほうが詳しいかもしれない。問題にすべきは，どうやってユーザーインタフェイスのよいゲームを作るかだ。そのためのポイントを押さえてみたい。

習慣

慣れ親しんだものと違う動作をするものを受け入れるのは困難だ。たとえば，ジョイスティックの左右方向が通常と逆のスクロールシューティングゲームがあったとしよう。少々凝った作りになっているからといって触る気になるだろうか？

私の知り合いの話をしよう。彼は，最近珍しくなった右手でジョイスティックを操る人である。どうしてゲーセンのスティックが左なのかは詳しく知らないが，右手で100円玉を入れさせるためではないかと推測する。右手のほうが連射能力が高いから左手でスティックを操作するというのはもっともらしいが，昔のゲームでは画面上に出ている弾数の制限がきつかったので連射は必要なかったはずなのだ。

まあ，昔からテンキーで自機を操作するのがパソコンの作法だった。つまり右手で移動，左手で攻撃だ。これにならい，初期のジョイスティックはどれも右手で扱うようにもできていた。

右手でスティックを使っていた彼にも，左手に持ち換えようとした時期があったようだ。しかし，無理してでも右手のほうが楽だと気づいてしまったらしい。愛用しているのはかなり修理と改造の加わったゼビウススティックだ。

ほっておくと突然あらぬ方向に走り出すスティックをねじ伏せてグラディウスやイメージファイトをクリアしてきた彼にとってやわなスティックはあわない。

もっとも有名なジョイスティック，電波新聞社のXE1-PROを使うとき，彼の両手はスティックの前で交差している。堅牢をもって知られるこのスティックも「メルヘンメイズ」においてセンタリングのきかない姿に変わっていった（新品が2台破壊された)。

こうして彼はもはや左手でスティックをコントロールできない体になった。染みついた習慣というのは簡単に改めることは不可能なのだ。

話が大きく横道にそれてしまった。

ユーザーインタフェイスに決定的なものは現れていない。ならば，それがよい習慣でないとしてもまったく無視することはできない。まして，これまで一定の評価を得ているもの，常識として広く受け入れられているものであればなおさらであろう。

しかし，世の中には「斬新さ」というものを勘違いしてか，故意にそういったものと違った動作をさせる人もいる。特にアマチュアに多いように思う。確かに，ゲームの進行は従来のものと変わってくる。しかし，それ以外にどんなメリットを生み出しているかというとかなり疑問がある。よほ

ゼビウススティック

X1など，パソコン用ゼビウスに同梱されていたジョイスティックのこと。四角いプラスティック製で初期のクリーム色，ブラックさらにゴールドの3種類があった。裏面のスイッチを切り替えて右手用にも左手用にも設定できる。X1ユーザーのあいだでは圧倒的なシェアを持っていた。その昔，祝一平氏がゼビウススティック買いますという告知を出したのを始め，いまだに愛用する人も多い。

その魅力はというと，右手でも左手でも操作できることもさることながら，スティックの遊びが非常に少ないことだ。中央にセンタリングした位置から，左右が認識される位置までの角度が浅く，さらに完全に倒し込んでも移動量は少ない。これは完全に倒し込んだ位置から逆方向に走る際にも機敏で無駄のない動作が可能だということを表している。腕を使ってガチャガチャスティックを動かすのと手首に力を入れるだけですむのとでは効率が違う。

頑丈でもある。ジョイスティックは製品の性格上，激しい振動などは当たり前，状況によっては床に叩きつけられるなどの衝撃に耐えねばならない。これらは使用状況から必然的に求められる特性といえよう。

もはや入手は困難である。激しい使用状況では接点部のゴムに亀裂が生じ，接点が入ったままになるという動作不良を起こす。定期的に接点の角度を変えたり，トリガ部のものと入れ替えるなどのメンテナンスは欠かせない。ジョイパッドのなかには似た構造の部品を使用しているものがあるので，臓器移植するのもよいだろう。

どのことがないかぎり，結局は改悪としか見られないのではないだろうか。

というわけで，SX-WINDOWのスクロールバーはアントニオ猪木が泣いて頼んでも受け入れることができない。

バランス

ユーザーインタフェイスとはなにか？

コンピュータの処理を進行させるために，どうしても人間の判断が必要になることがある。そんなときにお伺いを立てるところであり，その操作の結果をユーザーに伝えるための「工夫」のことだ。

もともと計算屋であるコンピュータにシステム全体を効率よく使うようにさせるためのものが「オペレーティングシステム（OS)」というプログラムだ。OSはいろんなものを管理する。

ディスクは無駄なく使われ，メモリやCPUタイムもうまくやりくりされる。こうしたコンピュータ関係の「資源」のうち，「人間」の面倒をみるのがユーザーインタフェイスの部分だ。

一般的に「目で見てわかりやすいもの」が喜ばれている。しかし，ゲームにおいてはプレイヤーの行動は画面上のキャラクターに反映され，すべてはビジュアルに進行する。“What You See is What You Get.”がいとも簡単に実現されているのだ。

必要なのは別次元での操作性だ。

たとえば，どんなに優れた表計算ソフトがあっても，データが入力できなければ使いものにならない。扱いづらいということはシステム全体の効率を著しく下げてしまう。結果としてよいユーザーインタフェイスは処理を進めるうえでの効率よい設計というものを実現するものということになる。

誰にも進めないシューティングゲームはゲームバランスが悪いと同時にユーザーインタフェイスが悪いといわれてもしかたがない。強力な敵に対処するには相応に強力なユーザーインタフェイスが必要だ。それは弾の雨の中でも確実に反応するスティックの応答性でも，攻撃を支援するオプション類でもかまわない。

ただし，オプションに頼りすぎているとバランスを失いがちだ。最強の装備を失ったときシューティングゲームの本質的な操作性が剥き出しにされる。グラディウスやイメージファイトなど，よくできたゲームにはオプションを失ってからでも戦う術が残されている。

ゲームにおける処理の進め方，それはゲームデザインと同義だ。ゲーム進行中はこれがすなわちユーザーインタフェイスとなりうる。結局はそういった心構えがゲーム全体の作りに反映されるといってもいい。

たとえば“Yet Another Column”ではゲームバランスを取るためにそれなりの労力が費やされた。テストプレイのたびにゲームバランス用の定数テーブルは原形をとどめないほど書き換えられ，得点計算部も修整されている。原作者すら正確な得点計算アルゴリズムを知らないはずだ。

“Yet Another Column”開発時の基本的なコンセプトを紹介しよう。それは短時間で終わり，アクション性が高く，技術と得点が結びつくということだ。末長く遊ぶためにも当初よりスコアファイルは必須と考えられていた。

ルールについてはアーケード至上主義の人には評判がよくない。確かに落としてからジタバタできるようなアマちゃんテトリスに慣れた人にはつらいかもしれない。パソコン版（ただしPC版やMacintosh版）のテトリスやコラムスは一瞬の判断力がゲームのすべてだが，アーケードではそれに加えて小手先の技術がものをいう。操作性の違いがゲームの本質を変えているといってもいいだろう。

実際，ユーザーインタフェイスはゲームを変えうる。あとちょっとユーザーインタフェイスが違えば，格段によくなるのに……と思うゲームもしばしば目にする。アーケードや海外の作品の移植では特に露骨に表れる。ゲームをして楽しいのはそのゲームに思い入れを持つからだ。確かにゲーム内容は同じなのだが，移植する人がそのゲームの本質をとらえていないとわかったときはやはり悲しいものである。

最後に

反論する方もいるかもしれないが，パソコンユーザーはユーザーインタフェイスについて寛容である。

アーケードフリークにとってはキャラクタと弾の関係は1ドット単位であり，ジョイスティックは絶対のものだった。しかし，パソコンではキャラクタはそれまで1ドット単位で動いていたゲームなんてそうそうはなかったのだ。

アーケードフリークに評判のいい移植ゲームはあまり聞いたことがない。せいぜいバブルボブルくらいだと思う。当たり前のことだが，その道の通ほど意見は厳しい。○○○蛇やR○○○○ならともかく，グラディウスやドラスピ，イメージファイト，サイバリオンでもかなりボロボロにいわれる。ま，これはゲーマー側の事情だ。

ゲームをするときプレイヤーとゲームデザイナーは対等な関係にある。プレイヤーとしてはゲームの都合を押しつけられるのは嫌だ。ゲームを作る側にすればわがままなプレイヤーは嫌だ。対話の接点，すなわちユーザーインタフェイスのガイドラインを確立することが重要となる。これまでも述べたように習慣に則しバランスの基準となるものを。

1つひとつのゲームが違う以上，それを得るための方法はテストプレイを徹底的に行うこと以外にありはしない。

簡単なこと，ではない。「不都合だとわかったら作り直すことができるか？」簡単なことではない。しかし可能なことだと思う。単にバグを見つけるためのテストプレイではなく，ゲームデザインの一環としてのテストプレイ。

もちろん，このときはユーザーインタフェイスについて寛容であってはならない。

日本のパソコンゲームは長いあいだPC-8801で実現できるもの，という十字架を背負っていた。ソフトハウスの人には失礼だが，その範囲に収めざるをえなかったというよりは，それ以上のものを想像できなかったといっていい（具体例を伴う反論には喜んで謝罪いたします)。実際，各ソフトハウスは必死で16ビットへの移行を行っているように思われる。ま，それでも「PC-9801で実現できるもの」でしかない。規模は変わっても発想は変わらない。結果としてハードウェアに頼りきるという安直なところに落ち着いてしまう。

AV面の強化は行われても，本来もっともハードウェアパワーを必要とするユーザーインタフェイス部分が見直されたことはどのくらいあるだろうか。

複雑化するゲームを簡単に見せる。水面下の白鳥の足のようにユーザーインタフェイスは戦い続けなくてはならない。

＊　＊　＊

よいものを作るためには。

1にも2にも，テストプレイ，テストプレイ，テストプレイ。ビジネスソフトなどならバージョンアップされるたびに速くなったり，遅くなったり，機能が増えたり，操作性がよくなったり，バグが増えたりする。ゲームソフトにもバージョンアップというのがあっていいと思う。

いや，ないというのがすでに間違いだったのではないだろうか。

ハードウェアからのアプローチ

資源の賢い運用を

Tan Akihiko 丹 明彦

コンピュータでゲームを作る以上，ハードウェアの制約を受けるのは当然ですね。でも，それらをいかにしてプレイヤーに感じさせないかという工夫も，よいユーザーインタフェイスを実現するためのポイントではないでしょうか。

私たちはパーソナルコンピュータで日々プログラミングにいそしみ，時にはゲームに打ち興じている(両者に費やす時間の比率についてここでは言及しないことにしよう)。パーソナルコンピュータというからには個人で購入できる程度の価格である。当然，その価格に見合った性能を持っている。もちろん，本当に価格に見合っているかどうかは機種によって異なる。

ともあれパーソナルコンピュータの性能には限界がある。しかしその限界を感じさせてしまうゲームを作りだすようでは本物のゲームプログラマではない。というより，限界をゲーマーにしわ寄せしてしまうようではゲームプログラマとして失格なのだ。

ゲームであるからには，ある程度まともに動いてくれないと困るわけだ。ここいらがビジネスソフトなどと比べて極度にシビアな部分である。「まともに動く」ことの定義は難しいが，とりあえず臨場感のある，というくらいのことにしておこう。パーソナルコンピュータの現実的な制約に引っ掛かっているとゲーマーに感じさせないこと，そしてそれをもとにしてゲーマーにストレスを与えてしまわないこと，が重要なのだ。

●要点

・ゲーマーは楽しくゲームがしたい。

・ゲーマーは臨場感のあるゲームを求めている。

・パーソナルコンピュータの処理速度は限られている。処理能力は主に速度のことを指す。

・解像度などの表示能力や発色数，音源の種類や発音数などにも機種によって制限がある。

・ゲーマーはゲームをするときまでそうした制限にとっつかまっていたいとは思わない。

・多少プログラミングの心得があるゲーマーは，愛機の性能に限界があるということを思い知らされるようなゲームをやりたいとは思わない。またそういう限界があるからといってしかたないとは思いたくない。

・優れたゲームプログラマはそれらの制限を上手にプログラミングテクニックで押さえ込み，決してゲーマーにそれをしわ寄せしない。

*

プレイヤーの意志を的確に計算機に伝え，行動の結果を的確に，またダイレクトにプレイヤーに戻す。これができていないゲームは，たいていやっていてつまらない。いいゲームは，ゲームデザインもさることながら，こうした技術的な面にも配慮を忘れない。たとえ面クリアできなくても，たとえエンディングを見ることができなくても，動かしているだけで面白いゲームは確かにある。

ゲームのプログラミングは，パーソナルコンピュータのハードウェアを最大限に駆使する一種の総合技術である。ゲームによっては，通信に使われるはずのRS-232Cポートでさえも対戦ゲームに使われる。手持ちのハードウェアをどのように使うかに，設計者のセンスが光る(月並みながら，結局はセンスの話になってしまうのだが)。

入力のおはなし

たとえばウィンドウシステムにマウスは必須アイテムである。逆にテキストエディタをマウスだけで使えというのもナンセンスであろう。どんな環境にもそれに合ったデバイスが存在する。ゲームだってそうだ。

さて，わがX68000のゲームを見渡せば，つながっているほとんどの装置はどこかでゲームに使われている。プレイヤーの手からX68000に意志を伝える道具，難しくいえば入力デバイス。ここでひととおりまとめておこう。

入力は主にデジタル的なものとアナログ的なもの，の2つに分かれる。もっともこれはプレイヤー側からの印象で，計算機にとってはどれもデジタルではある。アナログといっても多段階デジタルにすぎない。ま，重箱の隅をつっつくようなことばかりいってると嫌われるか。デジタル的なものにキーボードとジョイスティック，アナログ的なものにマウス，トラックボール，それにアナログジョイスティックがある。

●キーボード

当然のように標準でついてくる(というと問題がありそうだが……)。テキストをベースとして進行するゲームやアクションでも操作系統が複雑なゲームはよくキーボードを使わせる。しかしゲームにおけるキーボードはなにも文字を入力するためだけにあるわけではない。

意外な用途として，マウスなどを使うより正確に動かすためにキーボードが使われることがある。一本橋などのように，どうしてもまっすぐ行きたい，行かないと落っこちてしまう……，そういうときは，テンキーなどを使って正確に移動方向を定める，デジタル的であるところを買われるわけだ，ただ，当然のことながら，テンキーだと上下左右またはその組み合わせでしか方向が決められないので，90°単位や45°単位といった動きは得意でも，30°の方向に動こうとすると，ほかのどの入力デバイスよりも正確にしにくくなってしまうという笑えない事態に陥る。これはジョイスティックでも同じこと。テンキーやジョイスティックを使わせるゲームの舞台は，たいてい直交座標(要するに縦と横しかない)である。

●マウス

標準でついていたことの意味は大きい。本当に大きい。しまいにはアフターバーナーのような奇抜な使い方まで登場した(マウスが操縦レバーのようにふるまう)。

マウスは本来ポインティングデバイスである。画面に出てくるマウスカーソル(矢印)が，机の上のマウスの動くとおりに動く。手の動きに合わせて画面上のある一点を指示できる，それがポインティングデバイスである。

ゲームで，マウスをポインティングデバイスとして正直に使う作法もある。多くのアドベンチャーゲームやロールプレイングゲームがそうだ。昨年度のパソコンゲーム界を席巻したダンジョン・マスター，ポピュラス，シムシティーもこれである。ゲーム画面の中のいろいろな物体をポイントするためにマウスが使われている。

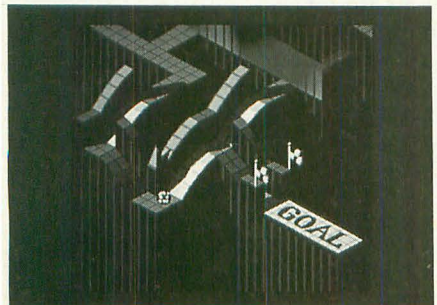
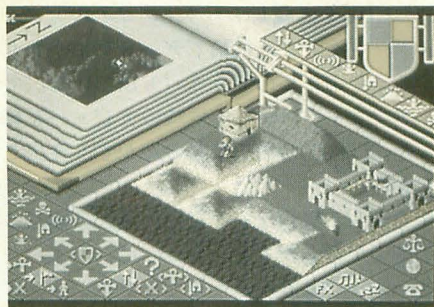
ほかにも，これらのゲームにはゲーム画面の外にアイコンがいくつもある。それらのアイコンをポイントし，機能させるためにもマウスが使われている。比較的複雑な操作になりがちな操作系統をキーボードに割り当てずにすませるにはこの方法が優れている。

余談になるが，ここで挙げた3つのゲームは，このアイコンの配置にセンスが感じられる。思考の流れに逆らわないとでもいおうか。また私見ながら，ポピュラスは直交座標から脱していながら(いわゆるクォータービュー)少しも使いにくくなっていないし，またアイコンから文字を極力排除して，わかりやすい絵を使っている点でデザイン的にも秀逸である。労多いわりには報われることの少ない日本語訳が入る余地がないことも得している。

今後出てくるゲームも，これら名作のシステムを見習って使える機能を画面上のアイコンに割り振るのはいい。いいのだが，よく練りもせずに下品に配置したものは目も当てられない。精進を怠ってはならぬ。

*

マウスもまた，アクションゲームでは多少風変わりな用途に使われている。アナログジョイスティックの代わりとして使うのである。先に述べたように，アフターバーナーがそうで，機体の傾きをマウスでコントロールできるというのは衝撃的ですらあった。

マウスがアクションゲームで使われる例としては，アルカノイドやシャッフルパックカフェを挙げることができる。これはマウスのポインティングデバイスとしての特

トラックボールの特性を生かした作品として代表的なマーブルマッドネス

画面のデザインが素晴らしいポピュラス。アイコンの使い方がまた見事だ

性を生かしたものであろう。狙ったところにパドルを持っていく，つまりパドルはマウスカーソルそのものなのだ。しかもアナログ（?）入力であるから，マウスを動かす勢いがパドルの移動速度に反映されるのだ。

マウスが激しい動きに向かないというのは，どうやら先入観にすぎないらしい。

●ジョイスティック

Xシリーズが伝統的にジョイスティックポートを本体につけているためか，ジョイスティックはほとんど標準となった観がある。

いまさら説明するまでもないと思うが，ジョイスティックの動作は上下左右およびその組み合わせで8方向を指定することと，ひとつまたは2つ(それ以上)のボタンを押すことである。

現時点ではもっともポピュラーなゲーム用入力デバイスである。X68000に移植されるアーケードゲームのほとんどは，ジョイスティックで操作するものである。ただ，ジョイスティックの動作はキーボードで簡単に代用できるので(たとえばテンキーとOPT.1, OPT.2キーなど)，それもサポートしているのがほとんどである。

X68000のジョイスティックポートは2つあるので，2本のジョイスティックをつないで2人用のゲームにするケースも多い。

また，多くのジョイスティックはスクロールシューティングゲームに使われるためか，連射を肩代わりしてくれる回路を内蔵している。

●トラックボール

箱の上面にボールが埋め込まれていて，それを手で回転させるとその動きを検出して計算機に伝える装置。早い話がマウスをひっくり返してマウスの腹にあるボールを手で転がすのと似ている。マウスと同じ原理で動作し，X68000の場合はマウスポートに接続するのでソフト的にはマウスとほぼ同じ扱いになっている。が，ゲームのインタフェイスとしてはかなり異なる地位にあ

ポピュラスには日本語訳がいっさいない。手抜きといわれようが成功している例であることは確かる。

トラックボールはまさにアクションゲーム向きの入力デバイスである。ボールは大きく，可動部分も高速かつ滑らかに回転することを目的に設計されている。

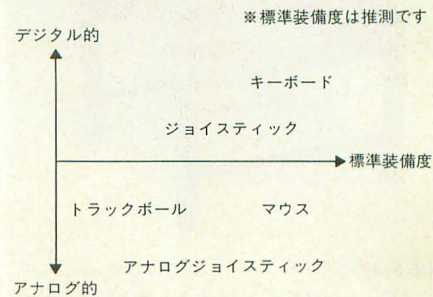
ポインティングデバイスとしての性格は薄い。マウスポートにつなげばマウスの代わりに使えるが，手の動きとマウスカーソルの動きが必ずしも一致しない。

トラックボールはそういう目的のためのデバイスではない。マウスが手の動きに合わせて位置を変えるためのデバイスであるのに対し，トラックボールは手の動きに合わせて速度と方向(2つ合わせて速度ベクトルだ!)を変えるためのデバイスである。

移動する方向がいかにもデジタルなもの(4方向や8方向)でない場合や，速度の微調整が必要な場合，特に速度と方向が刻一刻と変わるような場合にトラックボールは役に立つ。速く回せばキャラクタも速く，ゆっくり回せばゆっくりと，任意の狙った方向に動いていく。

サイバリオンやマーブルマッドネスはトラックボールを使うゲームの好例である。

ところで，X68000シリーズ(PROおよびPROII以外)のマウスはマウストラックボールと呼ばれている。マウスの蓋を開けるとマウスのボールが上面にも出ていて，トラックボールのように操作することができる。斬新だったのだが，残念ながらゲーム用のトラックボールとしては使いものにならない。机の上にマウスを転がすスペース

図1　入力デバイス

サイバースティックの発売と連動して話題となったアフターバーナー

のないときは重宝するのだが，ボールがマウス用で小さく，俊敏な動きができないので，ゲームに使うのはほぼ無理だろう。

トラックボールはシャープからも発売されているが（国内で売られているのはほとんどこのタイプだ），外国製のものに比べて使いづらい。持ちにくくてボタンが押しにくいのが主な理由である。もっとデザインに徹底的に凝ったものを作ってほしい。サイバースティックのようにシャープが自らデザインすればできるはずだ。

●アナログジョイスティック

そのサイバースティックの話。X68000のキーボード並みに大きな本体の上に操縦レバーとスロットルレバーを模した大きなレバーが立っている。通常のジョイスティックのような「レバーを倒した/倒してない」というデジタル的なものとは異なり，「レバーを大きく倒した/小さく倒した」というようなアナログ的な読み取り方をする。

前にもいったとおり，アナログといっても多段階デジタル。スティックの傾きを0～255の数値で表す。そのステップが十分細かいのでアナログに感じるというからくりになっている。アナログジョイスティッ

MENKURI。サイバースティックをはじめ，ほとんどのスティックが取りつけられる。大人にはちょっと小さいが，脚にキャスターをつけるといいかも

クはCPUを内蔵しており，ジョイスティックポートを介して通信でデータをやりとりする。そのためのドライバも出ている。

アフターバーナーと同時のデビューを飾った，X68000の周辺機器のなかでも突出した存在。

しかし不満がなくもない。最大の難点は，手応えが軽すぎるということだ。サイバースティックは誰がどう見てもアクションゲーム向けの入力装置だ。アクションゲームをしていれば熱くなるのがふつうの人の反応。そうなればレバーを力一杯倒すのは当然で，軽すぎるサイバースティックのレバーは苦もなく一杯に倒れてしまう。これではもはや“アナログ”ジョイスティックではない。「レバーを一杯に倒した/全然倒してない」では，ほとんどデジタルになってしまう。

あとは耐久性にやや難ありといったくらいか。目のつけどころはとてもいいのだから，もうひと頑張りといったところだろう。少々高くなってもかまわないから，しっかりしたものがほしい。おっと，こう書くといまのサイバースティックでも高くて買えない（24,800円）という声が返ってきそうだ。が，とんでもない。ゲームがしたくてX68000を買ったのなら同じ金額をジョイスティックに投入するくらいの気構えがほしい。

＊

以上，ここに挙げた入力機器も，ソフト次第でいろいろな使われ方をする。センスのいいもの悪いもの，ゲーマーに与える操作感覚も千差万別。

出力のおはなし

どんなゲームにもそれに合った，またはそれを実現するのにもっとも適した画面モードというものがある。

あるひとりのユーザー(つまり僕)の，快適なゲームに関する偏った見解を述べておこう。

・第一印象では音やグラフィックといった皮相の部分を見る。それがよければ，とりあえずは圧倒され，いいゲームのように感じる。

・その時期も過ぎ去り，しばらく遊んでゲームに慣れてきた頃には，初めどれほど感激したグラフィックや音にもそれほど感じなくなる。

・むしろ気になるのは速度や操作性といった部分に絞られてくる。

・その部分が腐っていると，あっという間に飽きられる。

・長く遊べるゲームの場合，こうした根底

一太郎文化とEmacs文化――プリンス・オブ・ペルシャ――

このところ海外ソフトで遊ぶ機会が多くなってとりあえずは喜ばしい。いつまでもそうではいけないとも思うが，けっこう目を覚まされた面はある。あちらの人は遊ぶことにも半端じゃないのか，いろいろなところで学ぶべきところがある。

それらの海外ソフトが国産マシンに移植されるとき，さまざまな「改良」を加えられることがある。

最近の作品ではプリンス・オブ・ペルシャがそうだ。PC-9801シリーズの高解像度の画面に合わせてグラフィックを全面的に描き変えて，見事なまでの変身を遂げた。まるで別のゲームのようだ。

個人的には，海外ゲームの移植は“卑屈なくらいに”オリジナルにべったりと移植すべきだと思っているので，こうした改良は歓迎できないのだが，別のゲームもしくはリメイク作品だと思うことにしている。

と，ここではこんな話をする気はなかったのだ。ここでいいたいのは海外版と移植版の操作体系の違いである。この原稿を書いている時点ではX68000版が出ていないので，こんな話をしてもしかたがないのかもしれないが……。

PC-9801版では(そしておそらくX68000版でも)，F1キーを押すとメニューが開く。そのメニューの中から目的の項目をカーソルキーで選び，リターンキーを押すとその項目の処理を始める。メニューは階層構造になっていて，次々とメニューウィンドウが開いていく。マルチウィンドウのようでいて全然マルチウィンドウではないのだが，それはこの際関係ない。

対して海外版(要するにAMIGA版だが，マニュアルを見る限りAppleIIやIBMでも同様である)では，コントロールキーで操作するのが基本である(むろん，通常のアクションゲームとしての操作はテンキー)。たとえばコントロールRでゲームをやめてタイトル画面に戻り，セーブしておいた面からはコントロールLでロードして続ける。そのほかいくつもあるが，ほとんどスクリーンエディタのノリである。それも機能をコマンドの頭文字のキーに割り当てるEmacsの作法だ。

これらを比べて，どちらがいいなどという不毛な議論をするつもりはない。ただ比べてみてとても面白く感じたというだけだ。ただ，使い慣れてから先，どちらが素早い操作ができるかはいうまでもないし，使い慣れないうちはどちらがわかりやすいかもわかりきったことだ。あとはこのゲームが長く遊べるゲームかどうかにかかっている。そして僕は，ここ2～3カ月というものこのゲームに夢中だった。

ときおり画面に現れる面数や残り時間などのメッセージにしても，PC-9801版は画面の真ん中にどかんと現れるのに対し，AMIGA版は画面下方の1行のエリアに慎ましく現れる(そう，Emacsのミニバッファのように……)。どちらがいいかはわからない。純粋に好みの問題だろう。

の部分がしっかりしている。そうしたゲームは表面的な部分もしっかりしていることが多い。

＊

こうしたことをふまえてよい画面モードに関する見解を述べておこう。

・解像度にこだわってはならない。
・色数にこだわってはならない。

結局みんな同じことをいっているのだが，つまりはそういうことだ。そして次のことも忘れてはならない。

・だからといって絵の質を落としてはならない。

画面モードとは？

順序が多少前後するが，ゲームでよく使われる画面モードをまとめておこう。ゲームの種類や性格によって，どういう画面モードを選ぶかが決まってくる。

●解像度

X68000の場合，解像度は768×512，512×512，256×256ドットとなっている。

そのゲームが文字，特に漢字を多用する場合は，なるべく読みやすくするために高い解像度を，アクションゲームなどでとにかくスピードが要求されるものだったりキャラクタの(見掛けの)大きさを重視したりする場合は低い解像度を選ぶ，というのが一般的な選択となるだろう。

●ディスプレイモード

X68000の場合ディスプレイの標準的な周波数は31kHz。ビデオ出力を考えるのであれば15kHzモードを使うことになる。画像の質感の違いからか，アーケードゲームの移植版でも15kHzモードはよく使われる。24kHzモードはPC-9801シリーズと画面の大きさを揃えたい場合などに，隠れモード的に用意されていることがある(24kHzモードを持たないディスプレイがあるので，24kHzモード固定のソフトウェアは作れない)。

●グラフィック画面のページ数

(1，2，4ページ)

●グラフィック画面の色数

(65536，256，16色)

たくさんの色を同時発色できるX68000。グラフィック画面だけで実に512Kバイトものメモリを占有している。そのグラフィックRAMを何枚かのページに分け，重ね合わせて表示することができる。4ページまで取れるこれらの画面はそれぞれ独立に動かすことができ，重ね合わせの順序も変えることができる。また，どれだけ緻密な背景を描いても，簡単にスクロールさせることができる。

ただし，ページを増やすほど同時に出せる色数も限られてくる。最大の4ページのモードのときは，同時に出せるのは16色だけ。ただし，その16色に割り当てる色は65536色の中から自由に選ぶことができる。その色の選び方に，グラフィック担当デザイナーのセンスが光るわけだ。

＊

ここでよい画面モードの話に戻ろう。

操作する者へストレスを与えない“気持ちのデザイン”をしっかりやればよい。というのではなんの解決にもならないので，もう少し画面周りに話を絞ってみよう。

・速度は速いほうがいい。

速すぎるならその余裕をもっと処理を充実させるほうに回すことができる。

・ディスクアクセスは少ないほうがよい。

人間は待たなくていいものに待たされると腹を立てるものだ。頻繁なディスクアクセスで理不尽に待たせたりするものはよくない。そういうゲームに限ってすぐゲームオーバーになってしまって，ゲームしている時間よりロードしている時間のほうが長かったりする。

＊

速度を落とさないためには，

・解像度は落とす。
・色数も増やさない。

となる。これはなぜか。解像度を落とすと，おなじ大きさのキャラクタでもグラフィック画面に書き込まなくてはならないデータの量が大幅に減る。ディスクへの格納スペースも減らすことができる。色数を減らすのもこれと同様の効果がある。

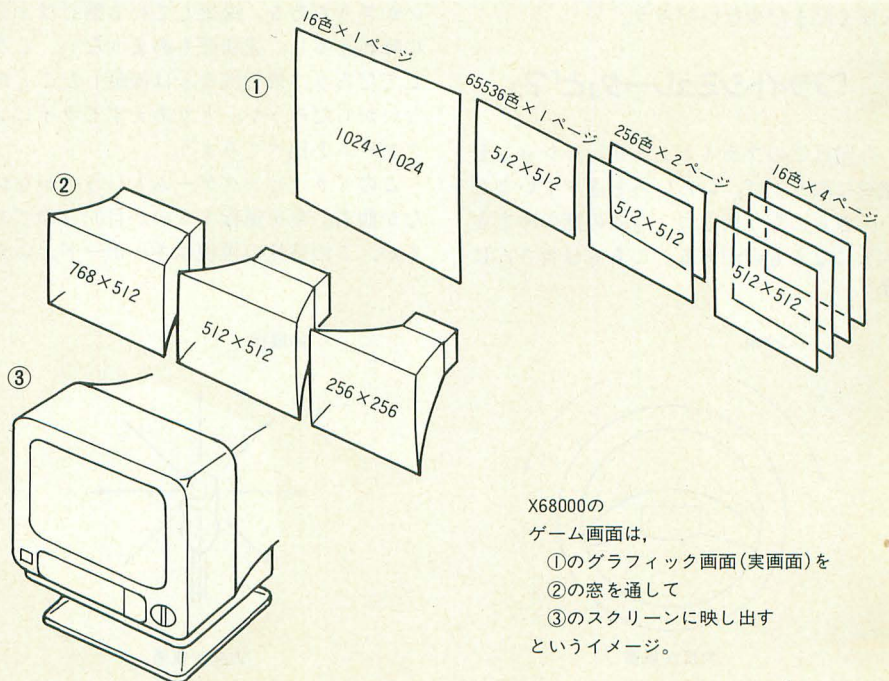
しかし，そのために絵が粗くて見られないものになっては本末転倒であろう。というわけで，

・少ないデータ量で良質の絵を描くことをこころがけるべきである。

という結論に達する。そのため絵師に要求される技術は相当なものだろうが，結局のところ解像度や色数に頼るよりは画力に頼った絵を使ったゲームのほうが長くやっていて心地よいものだ。

それよりもオンメモリにしたほうが喜ばれるのはいうまでもない。1Mバイトでオンメモリならいうことはないが，それがだめならせめてメモリを拡張している場合だけ処理を分けるようにする。そのためにもグラフィックのデータを下手にふくらまさないことは重要。

最後にちょいと話がそれるが，ぜひともいっておきたいことがある。4枚組だ，5枚組だとディスクの枚数の多さを誇ってはいけない。幸か不幸か，AMIGAはディスクドライブが標準で1基しかないので，1ドライブでそれほど不便なく遊べるというゲームが多いのだ。昨年の大ヒット作で，AMIGA出身のものを思い出してもらえばおわかりいただけるだろう。意味のない大作崇拝はそろそろやめにしたいものだ。どうせなら内容が深いものを，だな。

図2　主な画面モード

直感できなければ意味がない

模擬飛行と鍵盤

Akikawa Ryou 秋川　涼

日本ではさまざまなゲームの中で比較的マイナーな部類に属しているフライトシュミレータ。ここではあえてこのフライトシミュレータにスポットライトを当て、キーボードによる操作性について触れてみます。

今回のお話は……

前回（11月号）のゲーム特集ではWizardryを取り上げた。Wizardryはジャンル分けをするなら，RPG（ロールプレイングゲーム）ということになる。しかし，ある意味では人間の視点による3D（洞窟内探索）シミュレーションといういい方もできる。「Wizardry」の発展型ともいえる「ダンジョン・マスター」などを見るとそのことは明らかだろう。

さて，そのほかの3Dものというと皆さんどのようなものを真っ先に思い浮かべるだろうか。「スペースハリアー」や「アフターバーナー」などの前方突進型アクションゲーム，あるいは「アウトラン」などに代表されるドライビングゲームなどだろうか。

最近ちらほらと出てきたとはいえ，まだまだメジャーとはいえないフライトシミュレータ。これをいちばんに連想してしまう人はまだまだ少ないだろう。

「フライトシミュレータ」と「？」

今回はこのフライトシミュレーションを題材に取り上げ，いろいろと述べていきたい。さらにもうひとつ，今回の話の中で主人公になるものがある。こちらは皆さんにもたいへんなじみぶかいものであろう。それはキーボードである。

昔はアドベンチャーゲームはもちろん，アクションゲームだろうとなんであろうとキーボードでやっていた。それゆえ，よく手首や指が吊りそうになったものである。しかし，最近では入力装置なども多様化し，ことゲームに関してはジョイスティックやマウスの台頭により，キーボードは日陰に追いやられてしまったという感がある。X68000などはマウス標準装備だし，ジョイスティックの所有率もかなり高いものと思われる。

このように人気が薄れてきたキーボードだが，もうゲームをやるときにはうっとうしがられるだけなのだろうか。いやいや，あながちそうともいえない。ここでフライトシミュレータの登場なのである。

ほかのジャンルのゲームではなかなか使われなくなってきたキーボードではあるが，フライトシミュレータでは使っていて楽しいかぎりである。操縦している感じは十分に味わえるし，違和感もあまりない。どうしてだろう。飛行機などは操縦したことがないからだろうか。とりあえずドライビングゲームと比べてみよう。

このドライビングゲームというのがなかなか曲者。車を運転するのが目的なのであるが，この感覚の再現はキーボード，いやジョイスティックでも困難で，もはやハンドルをパソコンに取り付けるしか方法がないのではないかと思わせるほどだ。

一方，先ほども述べたようにフライトシミュレータでは飛行機の操縦感覚がかなりの部分，再現されているような気がする。ここで図1を見ていただきたい。左がハンドルの動き，右が操縦桿の動き。ハンドルの場合，右に動くか左に動くかよりも回転の角度が重要，操縦桿の場合は入れた方向が重要というのが一目瞭然であろう。

また，自動車はコースに沿った動きを要求されるという点で，ハンドルの傾きは非常に重要で，アナログの極致ともいえるだろう。飛行機の場合は動きの自由度は高く，入力がある程度デジタルでも許される。

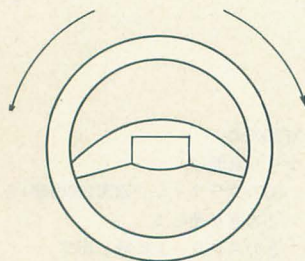
キーで左右

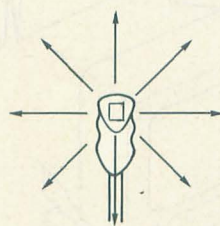
さて，実際にフライトシミュレータを操作するときには，どのキーがなにに使われるかでリアルな操作感覚が味わえるかどうかが違ってくる。まずは国内のフライトシミュレータを見てみよう。

X68000で日本純正のフライトシミュレータといえば，ほとんど「遊撃王II」のみである。「遊撃王II」はアナログジョイスティックにも対応しているので，本当はそちらの場合も考えなければならないが，ここではキーボードでの操作のみに絞らせていただく。

「遊撃王II」の場合，図2のような操作方法がとられる。使うキーの数も少ないし，特に難はないように思える。しかし，一点だけ少し使いづらく感じるのは，テンキー周り。1,3で左右に機体が傾き，2,5で機首が上下に向く（4,6では機体を傾けずに左右に曲がる）。この1,3,2,5というキー操作，慣れればなんてことはないのだろうが，やはり4,6,2,8のキー操作に親しんでいる指には戸惑いを感じる。慣れてからでも，熱中するうちに目的のキーとは違うキーを押

図1

ハンドル

角度が重要

操縦桿

方向が重要

してしまうこともある。個人的には少し問題ありといいたい。

どのキーがどの動きに

さて，それではどのようなキー配列がいいのかということになるが，実際にほかのフライトシミュレータでの例を図3に挙げたので見てほしい。海外製品ばかりなので，耳にしたことのないゲームも出てくるとは思うが，囲みで紹介しているのでそちらを参照のこと。

まずは，左右へのロール，機首の上下などの飛行機の動きである。これはテンキー（4,6,2,8）を使う「FALCON」，「DRAGON STRIKE」などのグループ，そして，カーソルキーを使う「F29」，「STRIKE ACES」などのグループの2つに分かれる。例外はほとんどない。まあ，定石どおりということだろう。テンキーのないコンピュータのためにカーソルキーでもできるようになっているのかもしれないが，望ましいのはテンキーのほうだろう。カーソルキーが十字型ではないコンピュータもあるからだ（AMIGAもそう）。そのうえ，周りに違うキーなどがある場合にはますますミスが多くなる。X68000の場合は十字型だからカーソルキーでもいいかもしれないが，次に述べるスロットルとの絡みがあるので，やはりテンキーがいいと思われる。もちろん，両方使えるのがいちばんいいのは決まっている。

スロットルが問題

次にスロットル調整であるが，これはアナログである。さて，どういうふうに再現すればいいのか。いちばんに思いつくのは，2つのキーを使って増減させる方法。

「遊撃王II」の場合がこれで，X,C，あるいは7,8キーで増減するようになっている。まあまあいい割り当てである。右手を方向

図2　遊撃王II

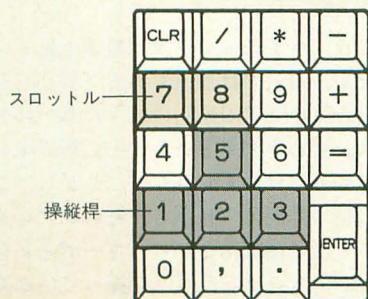
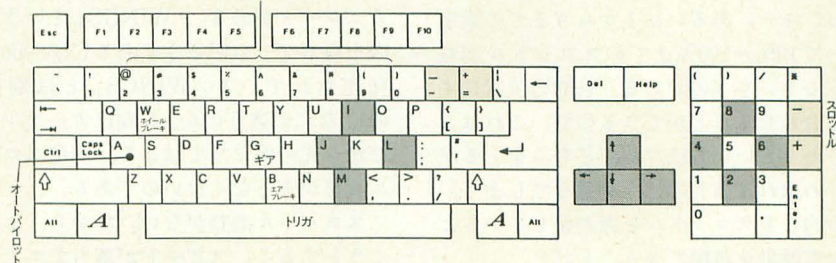



図3

●F29

●FALCON

●F/A-18 INTERCEPTOR

●STRIKE ACES

●DRAGON STRIKE

に専念させたいならX,C, 左手が動きの操作以外のこと（武器発射など）で精一杯なら7,8のキーを使えばいいのだ。

ほかのゲームの場合にもこの方法は使われている。しかし，キーは上の2パターンのどちらでもなく，+,－キーを使う。これはテンキーの近くにも，また（テンキーではない）数字キーのとなりにもあるので，X,Cと7,8と同じく右手でも左手でも操作できる。+,－キーを使うことにはもうひとつ，大きなメリットがある。そう，視覚的にわかりやすいのである。

例外としては，「STRIKE ACES」，「F/A-18」のようにF1からF10までのファンクションキー，あるいは1から9までの数字キーで10%～100%までのスロットル調整をするというものがある。目的の位置にすぐに合わせることができるので，これはこれでわりといい方法だが，どちらをとるかといわれれば前者のほうを選んでしまう。気分的にもスロットルを微調整しているような雰囲気を再現できることだし。

そのほか

そのほかの操作はあまり使わないので，+,－キーと同じく視覚的にわかりやすいほうがいい。ほとんどのゲームでそうなっているようにGEAR（車輪の出し入れ），BRAKE，FLARE，CHAFFがそれぞれG,B,F,Cのキーというパターン。これだとパニックに陥ったときも，即座に判断できる。

武器選択はリターンキー，ゲームの一時停止はP，エスケープキーはイジェクト，視点切り替えはファンクションキーがいい。もちろん，使うキーはほかにもいろいろとあるが，さしあたって使うのはこれぐらいなので，そのへんは適当でいいだろう。

ぜひとも統一化を

こういうふうにキー操作はある程度のパターンに統一されているので，フライトシミュレータを何本もやっていると，説明書を読まなくても何回か試すだけで，飛ぶことができてしまう（また，このことで周りで見ている人に対し，優越感を感じることができる）。しかし，実は飛ぶことができてしまうではなくて，何回か試せば飛べなくては困ると思うのである。

説明書を読まなければ，何回やっても飛ぶことさえできないとか，キーボードにオーバーレイのようなものをかぶせないとプレイできないキー配列，というのは迷惑なだけでなんのメリットもない。

昔のフライトシミュレータファンは，複雑なキー配列を使って，すごく現実的な飛行感覚を味わうこと，またそれを使いこなすことに，快感を覚えていたのかもしれないが，いまはもっとメジャーな分野になっていくためにも，より簡単でわかりやすいキー操作で気持ちよく飛ぶことが重要であると思う。

例外中の例外

ここまでに述べたものと，まったく異なる色合いを持った操作系を持つフライトシミュレータがある。「WINGS」という複葉機を操るゲームがそうである（X68000用に発売されている「WINGS」とは別のもの）。なにが違うのか。方向はカーソルキーだから先の例に当てはまる。実はスロットル調整がまったくないのである。

スロットル調整がない。だから，上昇しようとすると，スピードが落ちてエンストをよく起こす。下降すればスピードがのる。それだけである。使うキーはもうひとつだけで，機銃のトリガのみ。これならジョイスティックでもできる。離陸や着陸のときはどうすればいいのか，と思う人もいるだろう。しかし，「WINGS」はそのあたりがうまく省略されたゲームなのである。

「WINGS」はシネマウェアシリーズのうちの1作であるから，ただのフライトシミュレータではなくストーリー性を持っている。出撃の前には映画のような音楽が流れ，自分の書いた日記が画面に現れる。その日記を読み終わると，離陸のアニメーションシーンに移る。このあと，状況説明が文字で（無声映画の間に入る説明のようなかたちで）表示され，それから抜けるともう自分は大空にはばたいている，という感じなのである。ミッションを完遂するとまた文字が出て，日記に戻る。

実にうまい演出だと思う。要するに，フライトシミュレータをデフォルメしたゲームだといえるのだが，この演出のうまさからほかの本格的なフライトシミュレータと比べても遜色のない雰囲気，パイロットの気分を味わうことのできるゲームに仕上がっている。ジョイスティック1本だけで操作するにもかかわらずである。

キーに対する反応

ここまででキー入力のことを考えてきた。しかし，入力されてそれでオシマイというわけではない。「WINGS」の例でもわかるようにコンピュータの反応というのがあって，はじめて操作系のよしあしというものを感じることができるのだ。

反応は速ければ速いほうがいい。キー入力してしばらくしてから動き出す，というのでは予測した入力が必要になるし，第一快適といえる状態は見込めない。

また，回転したとき，地平線の回転の角速度は速いけど，滑らかな回転ではなく，10度とか30度とかの割合でカクカクと動くというのもある。これは全然よろしくない。多少回転が遅くとも，滑らかな回転，こちらのほうが見ていて気持ちいい。

最初のほうにも少し登場したが，アナログジョイスティックというものがある。ただのジョイスティックの場合だと，フライトシミュレータをプレイしようとするときキーボードを必ず併用しなければならない。しかし，アナログジョイスティックを使えば，それだけでプレイ可能となる（ボタンも多いから）。しかも，アナログだからスティックを倒した角度で，機体の傾き，スロットルの調整などを細かくできる。

とはいえ，現在のフライトシミュレータのレベルでは，まだアナログジョイスティックを使ってプレイすることに，あまりメリットはないような気がする。なぜなら，前述した画面上の反応がついてこられないからだ。アナログジョイスティックをすばやく動かしたときに，画面のほうもすばやく回転するなりしないと使っている意味がまったくない。

入力装置がいくら発達して多様な入力ができるようになったとしても，それに対してソフトウェアのほうがきちんと反応してくれないとうかばれない。現段階のフライトシミュレータではまだまだキーボードからの入力で十分である。しかし，そうかといってキーボードでやるのが面白くないというわけではなく，むしろ最初にいったようにそれはそれで面白い。

最後に

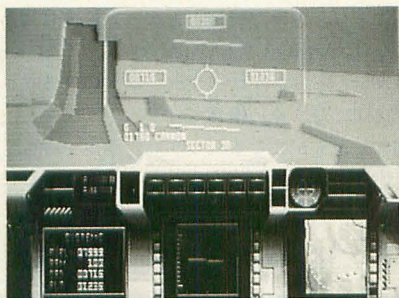
フライトシミュレータは数多く見ているのだが，最近では亜流である「WINGS」，「DRAGON STRIKE」のほうに惹かれてしまう。変わりダネということで新鮮味を感じているだけかもしれない。しかし，これらの持つ雰囲気の見事さ，それがプレイしている側に催眠術をかけ，キーボードを操縦桿，あるいはドラゴンを操っている手綱に感じさせている気がしてならない。

海外モノをちょっと紹介

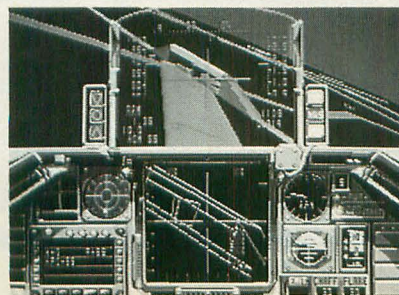
F29

文句なしに速い。しかも，カクカクした速さではもちろんなく，滑らかな回転も実現している。そのため，操縦中はグニャグニャ（？）した感覚が味わえる。機体は「F29」と「F19」の2種から選ぶことができる。

地形は遠くの物体は見えないが，近づくにつれ複雑な形状になっていき，目前にきたときにはかなり細かく描かれている。飛行できる地域は4種類。アメリカの砂漠の訓練場，大西洋上，中東，ヨーロッパ。ミッションはそれぞれに10ある。最初の訓練場のミッションでさえ，「道路を走っているコンボイ集団を攻撃せよ」とか，「線路を走っている列車を攻撃せよ」などと面白いものが揃っている。

しかし，速いせいで敵戦闘機が出てこないにもかかわらず，地面に墜落してゲームオーバーということも多い。あと，ディスク2枚組でわりと入れ替えがあるのが難点（プレイ中に読みにいくわけではないが）。これさえなければ，完璧にいちばん面白いフライトシミュレータといい切れるのに，残念である。オートパイロットもあったほうがよかっただろう。

FALCON

これもわりと速いほうだろう。だが，それよりもこのゲームの最大の売りは，前述の「F29」にはなかった「オートパイロットの賢さ」と思われる。なにしろドッグファイトをするのにも，このオートパイロットをオンにすれば，こちらはトリガを引くだけでいいのだ。これならフライトシミュレータが苦手な人でもすぐに遊ぶことができるだろう。

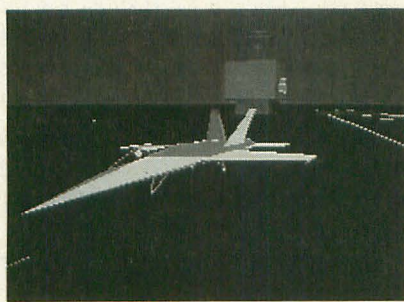
飛行できる地域はひとつだけということになるが，別売のミッションディスクが2種類ほど出ていて，こちらを使うと新しい地形，新しいミッションが楽しめる。コクピット周りの細かさ，視点の切り替えの豊富さという点も評価できる。ハードディスクにもインストール可能。まさに，親切設計という感じ。

おっと，もっと素晴らしいことがあった。なんと，もう1台AMIGAをつなぐと対戦が可能なのだ（Macintosh，ATARI STなどともつなげる）。ディスクにはプロテクトはかかっていないので（マニュアルプロテクトあり），この場合にもゲームディスク自体をもう1個買うなどということは不要。

F/A-18 INTERCEPTOR

このゲームは1988年に発売されたゲームで，現在では19.95ドルの安さで売られている。ベストセラーといったところか。

センスのよさは抜群で最新のゲームにも負けないものを持っている。まず，地図に表示されている4カ所の空港（空母も含む）から，ひとつを選ぶ。そして，「F-16」，「F/A-18」のどちらに搭乗するかを決める。すると，その空港がどんどんとズームアップされ，気づいたときには横から戦闘機を見ている画面になっている。ここまでの視点の移動が実にスムーズで，最初はかなり驚いた。

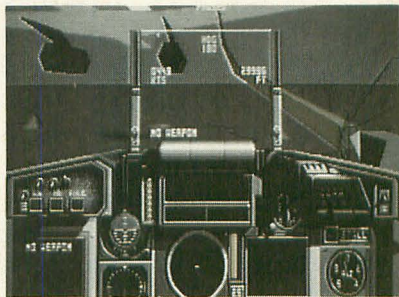
練習飛行というのがあるのも面白い。まずは，デモのような感じでいろいろなフライトテクニックを見せてくれるのだが，その次の段階に進むと自分の前を先生の戦闘機が飛び，それについていき先生の飛ぶとおりにマネしなさい，ということになるのである。

デモフライトも入っていて，空母からの離陸，敵戦闘機の撃墜，橋くぐり，そして着陸を見せてくれる。このデモを流しておくと環境ソフトのような雰囲気がある。

STRIKE ACES

スピードはそれほど速くないが，登場する戦闘機がとにかく豊富。自機では「F-15E」，「F-4E」，「F-111F」，「トーネイドIDS」，「AJ37」，「Mig-27」の6機，敵機では「F-14」，「F-16」，「F5E」，「Mig-29」，「Mig-31」，「ミラージュ2000」，「Su27」の7機が用意されている。もちろん，外見やコクピット，操作感覚などが変わる。

しかも，選択画面ではその戦闘機を3Dで表示して，ぐるぐる回して見せてくれる。また，地形のほうではMt.RUSHMOREの4人の大統領が彫られた岩山なども登場し，見る者を驚かせてくれる。

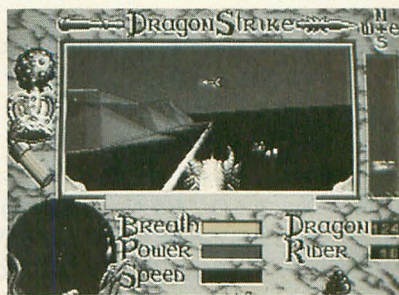
ミッションは隠密行動，戦術行動，戦略行動，攻撃行動の4つに分類される作戦がそれぞれ4つずつ用意されている。また，それに加えて任務を自作することもできる。

ミッションに入る前に内容がテキストと地図で説明されるのだが，このときに目標や任意の地点を3Dグラフィックで見ることができるのもうれしい。また，衛星視界，航路視界，友軍機からの視界ではズームイン/アウト，上下左右へのパンなどもサポートされている。

DRAGON STRIKE

なんと，このゲームはAD&Dシリーズ。なぜ，基本的にRPGであるAD&Dシリーズのひとつなのに，フライトシミュレータなのかというと，ドラゴンに乗って空を駆け回り，敵のドラゴンと戦っていくという内容だからである。

このゲーム，写真での見た目もあまりパッとしないし，期待はしていなかったのだが，プレイしてみると結構面白い。戦闘には飛び道具ももちろん使うのだが，基本的には槍を手にして相手に突進して1頭ずつやっつけていけばいいのだ。これが普通の空中戦と違っていて，なかなか気持ちいい。

ドラゴンのはばたく音が出力されるというのも，気持ちいい。相手のドラゴンをやっつければ，悲鳴をあげておちていくし，自分が死んだときは地面に叩きつけられる。このあたりの雰囲気のよさはなんともいえない。

ミッションも豊富で，ミッションに成功するたびごとに位が上がっていき，レベルアップするというところにRPGの要素がちらりと見えるのがご愛敬。グラフィックシーンもかなり多く入っている。

マウスで操る自動車レース
—Indianapolis 500—

ここ数年，世の中F1ばやりである。F1がどんどんミーハーになりつつある。しかるに，わがX68000にはまっとうな自動車レースシミュレーションがひとつも出ていない。誤解のないようにいっておくが，「まっとうな」ものが出ていないとはいったが，面白いものが出ていないとはいっていない。ただ，僕が面白いと思ったものが出ていないというだけだ(げっ，フォローになってない。……いいもん誤解されても)。物理法則をまったく無視しきった動きをしてみたり，2次元のキャラクタを拡大縮小しただけで3次元の遠近感でございといってみたり。酔いしれるような排気音が出るかと思えばそれもない(そういやスーパーハングオンでいちばんリアルな排気音は「自分が転倒したときの他車の排気音」だったなあ，転倒して頭にきてるときにだけ聞こえるいい音とは皮肉なことよ)。本当に車好きが作ったのかとまでいいたくなる(過激……)。

で，まっとうな，というのは，端的にいって3Dポリゴンものである。3Dポリゴンものであればいいというものでもないが，これだといいかげんな(もとい，よくできた近似をしている，ですな)ことをしにくいので，相対的に信頼度が高くなる。

1年前のゲーム特集でレースゲームのことを書いた。半年前のゲーム特集でフライトシミュレーションのことを書いた。そしていま，自動車レースシミュレーションのことを書く。最近になって感動ものの自動車レースシミュレーションに出会ったからだ。それがIndianapolis 500である。AMIGAやIBM PC (DynabookでもOK) で遊ぶことができる。

3Dポリゴン表示ながら異様に速い。排気音も，これがレーシングカーの排気音だ，といわんばかりの音が出る (AMIGA版)。自分より速いマシンが後ろから迫ってくるのがエンジン音でわかり，抜かれたあとはその音が前方へ遠ざかっていく。

絵と音がいいのはAMIGAゲームの常だが，それにただただ圧倒されている頃を過ぎると，このゲームの真価が見えてくる。

操作系は本当に単純である。AMIGA版はマウスを左右に動かせばハンドルを切り，マウスの2つのボタンでアクセルとブレーキ。

アクセルがアナログでなくていいのか？ これが不思議なことにいいのである。コースが単純なせいもあるのだろうが，全周回を通じてアクセルは全開である。それでゲームが成立するのか？ これまた不思議なことに成立するのである。コーナリングのタイミングがよければ見事に曲がりきるし，遅すぎるとコース脇に接触してタイムが落ちる。

下手でもこれくらいは楽しめる。そして上達してくるとこのゲームの奥の深さが見えてくる。僕は上手な人のプレイを見ていてこのゲームの奥の深さを垣間見た(情けない)。

デフォルトのチューニングはわりに甘いセッティングになっていて，初めて遊ぶ人でも周回が必ずできるようになっている。そこから先は，セッティングを煮詰めに煮詰めて100分の1秒単位でラップタイムを縮めることに熱中していくことになる。ダウンフォース，サスペンションやタイヤの固さ，そうしたセッティングを少し変えただけで車の挙動が如実に変わる。たとえば直線のスピードを伸ばすと，当然のように曲がりにくくなる。

セッティングを極限まで煮詰めて，速いけれど扱いにくくなった車で，アクセル全開のままコーナーに飛び込む。コーナリングのきっかけがちょっとでも遅れたらたちまちクラッシュする。タイヤのグリップが限界に近づくとタイヤが悲鳴を上げる。テールが振られるのもわかる。スリルに満ちたコーナーの立ち上がり。コースの幅をいっぱいに使っているので，コーナー出口付近では壁がぐんぐん迫ってくる。コーナーを脱出するのが早いか，曲がり切れずに接触するのが早いか。緊張の一瞬。

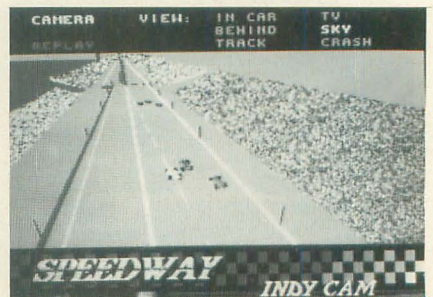
本当に速いセッティングのマシンでコーナーをきれいに曲がりきるのは針の穴を通すのに等しい技だ。限界走行を1時間も2時間も続けるF1のトップレーサーたちが神のようにも思えてくる。が，こうしてコースレコードを叩き出したときの達成感はなかなかのもの(らしい)。このゲームで遊んでいると，作者のクルマが好きで好きでしかたがないという思い入れをひしひしと感じてしまう。

操作が簡単でもその他のデザインで上級者でも楽しめるものが作れるという好例といえる。操作系は複雑に，本物そっくりに作ればそれでよいというものでもないようだ。愉快なのが，トラックボールで走ったときがもっともタイムが伸びるというところ。この手のゲームの多くはアナログデバイスへの対応がおまけ程度にしかなっていないが，これだけコーナリングが微妙でシビアなゲームになると，ハンドルさばきをトラックボールほど精密に表現できるものもない。

ゲームの構成が練習・予選・決勝に分かれていて練習ステージではこころゆくまで走れるし，失敗してやけになったら逆走してほかの車にぶつかって大事故も起こせるし，それをあとから再生していろんな視点から見ることもできる(クラッシュは本当に派手で，車体の破片が粉々になって飛び散る。妙なところまで凝っていて，クラッシュするのが楽しくなるくらいだ！)。

基本的にオンメモリで動く。さらに驚くべきことに，どんな機種でもゲーム速度は変わらない。AMIGAでもDynabookでもCompaqの386マシンでもコーナーリングのタイミングは同じだ。速い機種では表示がリアルに，動きが滑らかになる。こうしたツボを押さえた作りもあいまって，実にゲーム性の高い作品である。ぜひともX68000への移植を実現してほしいものだ。 (A.T.)

ピットインももちろんできる

レースは圧倒的な臨場感だ

いろんな視点から見れる。再現モードがすごい

マウスオペレーションの心

人に優しいって，なんだ？

Ogikubo Kei 荻窪　圭

ゲームに限らず，ユーザーインタフェイスを考えるうえで注目されるマウス。その使い方によって，ゲームは優しくも，意地悪くもなるのです。マウスを使ったプレイヤーとゲームのコミュニケーションのあり方を皆さんも考えてみてください。

わかりやすさの現状

Macintoshはパソコンではない。結局，そういう結論に達した。ではなにか。Macintoshは“Macintosh”というものである。少なくとも，エンドユーザーにとってはそうだ。私はMacintoshに関してはエンドユーザーに徹することにしたので，Macintoshがパソコンとは思っていない。もう少し補足すると，ジョブズはMacintoshというモノを作り，スカリーがそれにパソコンという要素を付け加えた，というほうが正しい。

これはなかなか面白い発見だった。Macintoshを嫌いな人は，Macintoshをパソコンと思うからいけないのだ。あれは，Macintoshというモノなのである。そう思えば，実に可愛い。その証拠に，立ち上げ時に“Welcome to Macintosh.”とメッセージを出す。Macintoshという環境へようこそってな意味だ。Macintoshは，パーソナルワープロやファミコンをパソコンとはいわないのと同様，パソコンではないと思う。

どうせだから，Macintoshからパソコンを思わせるメッセージはどんどん削るべきだ。たとえば，爆弾マーク(SX-WINDOWでいうドクロマーク)が出たときは，“システムエラーです”というメッセージが出る。これはよくない。“I am confused.”とすべきなのだ。ついでに，スペシャルというメニューは“その他”にしたほうがいい。せっかく独自の言葉をたくさん作ったのだから，そこまでいかねば，と思うのだ。さあ，君も，「小学生になろう」(笑)。

そんなわけで，今回の原稿は始まる。

いきなりだが，なかなか笑えるので，次の言葉を引用しよう。

「考えてみると，

人間のほうがパソコンの都合に合わせていること，

ずいぶんあったような気がします。

誰もがもっと自由にパソコンを操れるように。

誰もが情報を共有できるように。

……」

これは何かというと，マイクロソフト社の広告の一部である。もちろん，WINDOWS 3.0の広告だ。

いったいぜんたい，“人間のほうが何かに都合を合わせないこと”があるだろうか。パソコンに限った話ではない。人はすべての機械や環境や道具に都合を合わせて生きているのだ。そこのところを勘違いしてはいけない。そこにあるのは，“都合を合わせやすいかどうか”だけである。さらにいってしまえば，都合を合わせにくくても，“みんながそれに都合を合わせていれば，従って”きたのが人間である。

私はこれから延々とユーザーインタフェイスについて書くわけだが，その前にいっておきたかったのがこれだ。

もう少し賢明な人々はこう思っている。“都合を合わせないで使えるパソコン＝パソコンがない時代に行っていた所業の感覚で操作できること”。果たして，それは可能だろうか。パソコンの中という抽象的なワールドで人はどこまで自由にふるまえるものなのか。っていう難しい話をする気はあまりないけどね。

一番使いやすい入力デバイスは？

とりあえず，だらだらと並べてみよう。キーボード，マウス，ジョイスティック，タッチパネル，音声，ペンタッチ入力。どれがいいか。

タッチパネルとジョイスティックが筆頭だろう。タッチパネルは銀行のATMで，ジョイスティックはファミコンで使われている。人間にとって慣れやすいインタフェイスであることは確かだ。

そのあとに，ペンタッチ入力を挙げよう。続いてマウスと音声が同等レベルだろう。音声はなかなか期待されているようだが，今の時代においてはこのレベルだ。10年後はどうかわからない。

最後にキーボードがくる。

では，よい入力デバイスの条件を考えてみよう。

1)　初めの一歩が簡単
2)　視線の移動が少ない
3)　筋肉の動きが少ない
4)　単純な動作と複雑な動作の間で比べた場合，2)や3)を満たす

という具合だ。4)ってのはわかりにくいが，つまりは，単純な動作なら簡単に指定できるが，複雑な動作を指定しようと思ったら途端に視線の移動量や筋肉の動きが増えるのはよくないということだ。

異論のある人はあとで文句をいうように。私はない知恵をしぼって，考えている。

キーボードは1)でまず×だが，4)は○。2)はタッチタイピングができる場合に限って，成り立つ。3)には×をつけたい。

ジョイスティックは1)～3)は満たすが，4)が×である。ジョイスティックを使ったRPGを見ればわかる。メニューを開いて，カーソルを動かして，という動作が非常にうっとうしい。コマンドをパソコンに伝える場合，そのコマンドが多岐に渡れば渡るほど，労力を必要とするからだ。

タッチパネルというのはいい線いきそう

図1-1

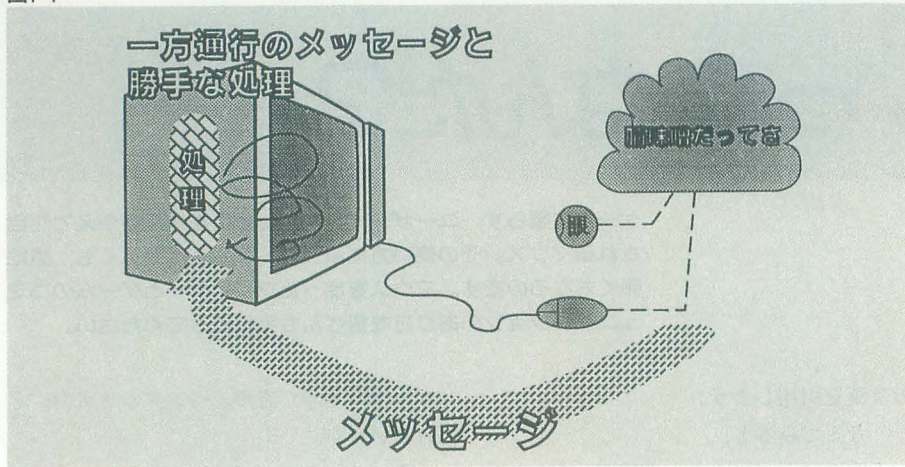
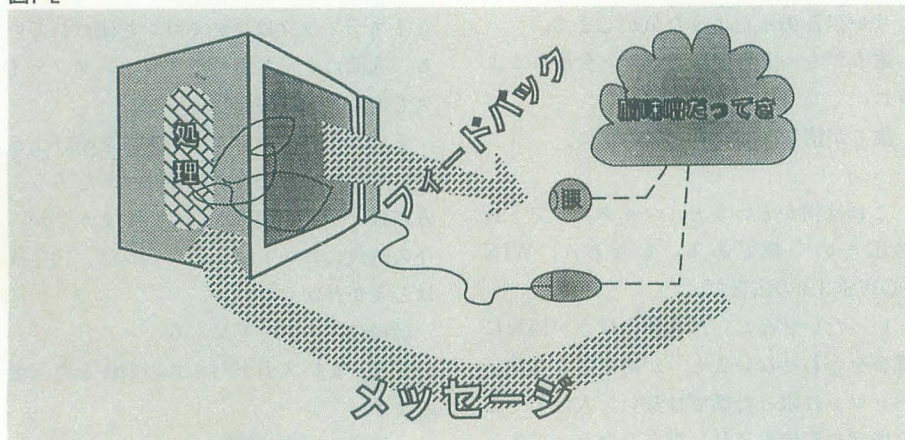



図1-2

だが，3)で×である。銀行のATMのCRTならともかく，ディスプレイは垂直面に立っている。腕を前に伸ばすのは疲れる。ペンタッチも同様。デスクトップパソコンには向かない。

音声というのを忘れていたが，命令の伝達に時間がかかるうえ，結局，コマンドを覚えねばならない。今は非現実的だ。私は留守電に向かってしゃべるだけでも緊張するのに，パソコンに向かってしゃべるなんてもってのほかだ，と思っているひとりである。

そういうわけで，マウスが生き残る。1)においてはタッチパネルやジョイスティックに譲るが，それ以外はなかなかいいバランスが取れていると思う。

ああ，長い前置きだった。予定調和な結果の見えていた文章を読んでくれた皆さま，ありがとうございます。

次は出力デバイスだが，何がいいか，てな話はもうしない。CRTディスプレイしかないのである。さっさと次にいこう。

猫をカン袋に押し込んで

「ポンと蹴りゃ，ニャンと鳴く」というわけで，これがパソコンの基本である。パソコンに対して何かメッセージを送ると，それに対して何か反応が返ってくる，ということだ。むっつかしい言葉でいうと，インタラクティブ。日本語では“対話的”というらしい。訳しても全然簡単にならないから困ったものだ。

パソコンにメッセージを送るというのも抽象的ないい方だな。このメッセージというのは，キーボードを叩いたり，マウスをクリックしたりという動作だ。

人間は常に相手の都合に合わせている生き物だ。これをいやらしい言葉でいうと，“気配り”となる。最近のはやり言葉でいうと，フレンドリーって感じ。

耳の聞こえない人に向かって大声で呼びかけるのは，フレンドリーでないというわけだ。

だから，Macintoshのマウスに向かって語りかけてもパソコンは動いてくれないからといって，文句をいうのも筋違いだ。

しかし，相手の都合に合わせたコミュニケーション手段を選択しているのに，それに応えてくれないのは悲しい。

街で話しかけた女の子に無視されたら悲しいのとよく似たものだ。

インタラクティブでないものの代表が留守番電話だ。一方的に要求しておいて，知らんふり。用件をいい終わったら“ありがとうございました”っていうとかだな，してほしいとは思わないか。だって，こちらはいいっ放し。うまくいったかどうかの確認さえとれない。だいたい，失敗したら録音のし直しくらいはしたいぞ。

技術的に無理なのに放置しているのだとしたら，人間側が“便利さに魂を売り渡している”のだとしか思えない。便利さと引き替えに何を譲歩するかという問題だな。もっとも，留守電に入れたメッセージに対する反応というものは，その相手からの連絡によってなされるものかもしれない。だとしたら，時間がかかりすぎである。

どうしてインタラクティブでないといけないか。人は自分の行動のフィードバックを常に欲しているからだ。

それを逆手にとったのがシューティングゲームにおける“硬い敵キャラ”だ（やっとゲームの話がはじまった）。

シューティングゲームの場合，フィードバックのレベルがいろいろとあって，もっとも低いレベルが，“ジョイスティックに力を入れると，その方向にスティックが倒れる”という“物理レベル”だ。その次のレベルがスティックを倒した方向に自機が動くとか，ボタンを押したら弾が出る，といった“1次反応レベル”だ。その次に，自分が壁に当たったら爆発するとか，弾が敵に当たったらそいつが雲散霧消するといった“2次反応レベル”がある。後者になればなるほど，そのゲーム独自の約束事が生きてくる。

さて，硬い敵キャラというのは，2次反応レベルのフィードバックに対する期待を裏切ることによって，プレイヤーを興奮させたのだな。

私はこういうのが嫌いだ。弾が当たったのなら，色が変わるなりなんなり，ダメー

ジを受けたことを示すフィードバックが欲しい。

マウスの場合だと，1次反応レベルは安定したマウスカーソル移動ということになる。

ユーザーインタフェイスの話から随分それてしまったが，まあ，ゲームというのはフィードバックをコントロールすることによって，成り立っているということをいいたかったわけである。

ユーザーが期待する動きって?

よくいわれるのが，ユーザーインタフェイスに対する次の意見だ。

「ユーザーがこうしたらこう動くだろうという期待に応えるのがいいユーザーインタフェイスである」

それはまあ，そのとおりである。が，ユーザーみんながみんな同じ動きに対して同じ期待をするわけがない。私は常に画面上の女の子をクリックするとその子が脱いでくれるという動きを期待するが，それが実現されているソフトを見たことはない。それはともかく，一番期待に反した動きをしてくれるのは，常に日本語FEPである。

しゃあないから，もう少し一般的な表現に直すことにする。

「ユーザーがこうしたらこう動くだろうという予想をしやすいのがいいユーザーインタフェイスである」

たとえば，一太郎という日本語ワープロソフトは，何度使っても，どのコマンドを実行すれば何が起きるか，という予想をたてられない。Lotus1-2-3というスプレッドシートのメニュー構造は，慣れるまで実に無駄な試行錯誤を要求する。どちらも予想しにくいのである。わざとマニュアルがないと使えないようにしたのでは，と思うくらいだ。

なのにここまで平然と普及しているのは，「ユーザーは予想することをやめ，受験勉強のように，ただ覚えることにした」からである。無駄な予想をして裏切られるくらいなら，はじめから期待せず，覚えてしまえ，ってなわけだ。こういうソフトは，自分で考えてソフトを使う人にはすこぶる不評だ。

うーん。不幸だ。っていうより，こういう受験勉強のような人生って，きっと退屈だと思う。

では，Macintoshのソフトは，というと，決して日本人には予想しやすいものではない。が，ひとつの機能を実行するためにいくつものキーを押してメニュー階層を深くまで降りていく必要があるDOSアプリケーションに比べ，プルダウンメニュー1発（あるいは2発）ですむから，試行錯誤してて腹が立ちにくいのだ。単純であり，なおかつ“Macintoshの世界”というワールドにおいては非常に整然としているため，非常に慣れやすい。さらに，ほとんどのソフトがアンドゥ機能を装備しているため，間違えたらチャイできる。そうこうしているうちに，Macintoshの動作は予想できるようになる（英語アプリケーションでさえも)。

ここで私はもうひとつ付け加えたい。「頻繁に使う動作は簡単なオペレーションで実現できること」というものだ。部屋を片づけるとき，一番上手な片づけ方は，部屋の中で自分が一番よくいる位置を中心にして，近いところから遠いところへ向かってよく使うものから順に置いていくことだ（あまり知られていないけど)。そうしないと，どんなにきれいに部屋を片づけても，1週間で元に戻ってしまうのだ。

ワープロでいえば，カット&ペースト，そしてファイル操作などが単純な操作で実現されるべきもの。たとえば，一太郎で文字列を削除しようとすると，“ESC,D,カーソルを始点へ持っていく，Return，カーソルを終点へ持っていく，Return”の6ステップが必要だ。始点の指定は省略できるがそれを抜いても4ステップ。Hyperwordなら，“XF1+カーソルキーで範囲指定，XF1+X ”の2ステップだ。この差は大きい。とっても大きい。一太郎というとすぐ“遅い”ということが問題となるが，致命的なのは遅さよりも，こうした操作体系なのだ。

こういう単純なキー操作と理路整然としたメニュー構造を同時に成り立たせる。おそらく，メニューと慣れたユーザーのためのショートカットキーとマウスの併用が妥協としてベターだろう。

どうも，今月はなかなかゲームの話にならないなあ。困った。困ったぞ。というわけで，話を変える。

期待と裏切りに揺れる乙女心

前に，シューティングゲームにおける，ユーザーの操作に対するフィードバックについて話した。

一般にユーザーインタフェイスと呼ばれるものは，物理レベルと1次反応レベルまでである。この2つはユーザーの期待を裏切ってはいけない。私がサイバースティックを使わないのは，物理レベルにおいて，レバーはもっと重くあるべきだ，という期待に応えてくれないからである。物理レベルに関しては，ゲーム側の責任ではない。

1次反応レベルに関しては，ゲーム側の責任だ。次の項からはちゃんと1次反応レベルの話をする。

2次反応レベルから上のn次反応レベルが難しいところだ。ユーザーが期待したとおりの反応しかないとつまらないゲームだし，常に期待に反した反応しかないと，わけのわからないゲームになってしまう。コラムスが面白いのは，そのあたりがあまりにも巧妙だからだ。期待を超えた連鎖反応がプレイヤーを悔しがらせ，興奮させる。

ユーザーの動作が2次反応でフィードバックされるだけでなく，3次や4次まで伸びていけば，奥が深いゲームと称してもらえるわけだ。ダンジョンマスターやポピュラスはそういった点で，実に凝っている。

予想しやすいユーザーインタフェイス

週刊誌レベルの記事を見ると，アイコンがもてはやされている。そのレベルがまたすごい。「絵文字は視覚的でわかりやすいから」だそうだ。笑止千万。

だって，絵文字だけでは表せないから文字が必要になったのではないか。パソコンだって，すべての操作が誰にでもわかる絵文字にできるほど単純であったら，誰も困らない。アイコン神話ってのはその程度のものだ。Macintoshなんてアイコンの数が増えすぎて，なにがなんだかわからない。ただ，使っていれば覚えてしまうのがアイコンであり，そうもいかないのがややこしい言葉だというのはいえる。人間の認識の問題だ。

だから，私はアイコンを否定するものではない。アイコンで十分な局面において，実に有効な手段だ。

ゲームというものがある。ここではマウスを使った話をする（つもりだった）ので，そういうゲームを思い浮かべていただきたい。ダンジョン・マスターやらポピュラスやらシムシティーやらねじ式やらリバーヒルの一連のシリーズやらノベルウェアの一連のシリーズやら提督の決断やらアートディンクの栄冠は君に各種マージャンゲームやらで，いろいろたくさんある。これからも増えるだろう。絶対に増える。

マウスは基本的に，画面の一部を指し示すことによって，その示した場所に応じた反応が返ってくることになっている。

マウスが指し示す場所の表現方法でいくつかパターンを見出せるだろう。ひとつはアイコンを使う方法で，もうひとつは文字によるメニューである。アイコン派はダンジョン・マスター，ポピュラス，シムシティーの外国勢。メニュー派はねじ式など和製シミュレーションやアドベンチャーに多い。RPGもマウスは使わないが，カーソルキーによるメニュー選択を使うケースが多い。

それに加えて，特定のエリアではマウスの動作がアイコンやメニューを介さずにゲームに影響を与えるシーンもある。ポピュラスの土地上げ下げや，ねじ式のダンジョンモード，ダンジョン・マスターの手のアイコンだ。

この3つに名前をつける。“アイコンオペレーション”と“メニューオペレーション”と“ダイレクトオペレーション”である。

人間にとってわかりやすいとは

人間がパソコンから受けるフィードバックは，視覚と聴覚しかない。キーボードが振動したり，ディスプレイから匂いがしたりするのもいいだろうが，ディスプレイから匂いがしたらたいてい故障である。

だいたい嗅覚が欲しいとはあまり思わない。ビープ臭なんてかぎたくない（でしょ？）。なおかつ，匂いは慣れやすいから，いつしか刺激ではなくなってしまうという欠点もある。ビープ触ってのも怖いな。ビープ音の代わりに電気ショックが伝わったりして，そうなったら，耐ショック絶縁手袋でもしなきゃ。

振動するというのは非常に嬉しいので，振動するマウスとかジョイスティックってのは作るべきだ。私なんか，銃が振動するというだけで，ビーストバスターズなんて好きだもんな。

さてさて，パソコンからのフィードバックのほとんどを視覚に頼っているのが現状だ。だから，視覚に頼りきったマウスオペレーションが神聖視される。

しかし，人はディスプレイを常に凝視しているわけではない。画面の隅に小さく“エラー”と出ても，気づかないことが多い。キャラクタベースのシステムではそれが普通だった。画面は常に分割され，分割されたそれぞれが役割を持ち，ユーザーはそれを覚えないと正しくフィードバックを受けられなかった。そこで，Macintoshはダイヤログという画面のど真中に現れてそれに答えるまで次へ進めないという機能を搭載した。画面の真ん中に飛び出たものに気づかないものはいない。特にキーボードオペレーションが初心者の視覚の多くをキーボードに奪われるのに対し，常にディスプレイに目を向けさせるマウスオペレーションでは効果があった。

ところが，いつの世にも怠慢する人はいるものである。怠慢な人々は，“キャラクタベースのソフトウェア”にそのままマウスを導入してしまったのである。ああ，無情。

入力機器がキーボードのテンキーやカーソルキーからマウスへ移っただけの，メニューオペレーション。クリックが有効なのはメニューの上にマウスカーソルがいるときだけであり，さらに，画面中に調べたいアイテムが見えているのに，メニューから「調べる」なんていう項目をクリックせねばならないもどかしさ。これではマウス対応とは呼べない。

昔，右ボタンを押したらソフトキーボードがぽよんと現れたゲームがあった。我々

「脳波！」

Mac World Expoへ行って，面白い玩具を見つけた。あれで128,000円程度なら安い買い物かもしれない。なにかっていうと，脳波を計る機械だ。

まず，トランスミッターのついたヘアバンドを頭にはめる。前頭部内側に3つのセンサーがあり，そいつが脳波を計るわけだな。

トランスミッターがあれば，レシーバーもある。レシーバーはMac World Expoであるから，当然Macintoshにつながっているわけだ。マックの画面には脳波が刻々と表示され続ける。どのあたりがα波でどのあたりがβ波か，というのはわかるから，画面を見ていればリラックスしているかどうかわかるという寸法らしい。

私がやってみたところ，「リラックスしてませんね。リラックスしてください。子供の頃のことなんかを思い出して」などといわれてしまった。ばかやろう。そう簡単にリラックスしてたまるか。こんなところでリラックスできるのは，ただの脳天気か無神経だけだ，と思ったが，黙っていた。私は気が弱いのだ。そういえば，リラックスするって，どういうことなのか未だによくわからないなあ。

で，リラックスはどうでもいい。問題はこのIBVAという機械だ。性能は？　と尋ねたら，数百万円の機械と同じくらいだ，といってたので，それをとりあえず信じてやるとしよう。ということは，Macintoshがあれば，たった128,000円でこんなオモチャが買えるのだ。

脳波コントロールのゲームだって，夢ではないぞ。フィードバックを3次元グラフではなく，ゲームの反応にしてやればいいのだ。ゲーマーの心理状態によって敵の動きが変わるシューティングとか，α波がある程度でないと面クリアできないシューティングとか，こっちのα波やβ波の状態がキャラクターに影響を与えるRPGなんて面白そうではないか。

達人になったら，脳波のコントロールだけでシューティングができたりして。心が乱れるとボロボロっていうのもいい。

Ramdom ELECTRONICS DESIGN社としてはアメリカで積極的に売っていきたいそうだが（日本での売り上げは期待していないようだ），やはりそういう機器はセルフコントロールの好きなアメリカで受けるんだろうな。

というわけで，マックがあれば，こんなものも使えるのだ。仕様がわかれば，X68000にだってつながるだろう。さすがに，X68000版は出さないの？　とは聞けなかった（答えはわかっているし）。

脳波によるパソコンのコントロールを真面目にやりたいとはまったく思わないが，遊びの一環としては興味がある。このIBVAは2つのトランスミッターを使えるそうだから，2人協力プレイもできるのだ。へっへっへ。こいつは面白い。

ちなみに，脳波でパソコンをコントロールしようなどと考える輩も多いようだが，止めたほうがいいと思う。なぜか。人間は自分の思考をそれほどコントロールできるものではないからだ。そこまで人間は単純じゃない。自分自身をコントロールしたいのなら話は別だけど，そういう訓練はしたくないしね。

セルフコントロールってのがビジネスマンの世界でははやっているらしいけど，バカバカしいことだと思うよ。自分で自分を洗脳するようなものだもん。

心なんてのは社会や環境でもまれて強くなったり弱くなったりあざとくなったりクレイジーになったりするもので，意志でコントロールしようなんて傲慢以外の何者でもないのさ。ってばさ。

は叫んだ。「おお，フルマウスオペレーションだ」。こういうのと変わらないではないか。

だからして，受け入れることはできない。マウスを使うなら使ったなりの出来ってえものがあるのである。

ひどいのがマージャンだ。せっかくマウスがあるのだから，ポップアップメニューから“チイ”とか“ポン”とか選ばせるのは貧困すぎる。“食いますか？”っていうメッセージが出て，マウスカーソルが手になる。そして，相手の捨て牌をクリックすると，ポン！　って叫ぶ（チイ！　でもいいけど）のがいい。ロンや暗カンはメニューでもいいけどね。リーチだって，リーチをメニューから選ぶより，自分のハコから千点棒をドラッグするほうを選ぶな，私は。多少面倒かもしれないけど。ゲームは気分で遊ぶもの，だ。

たとえばこんなユーザーインタフェイス

私が思うに，“マウスカーソルの動くすべての領域において，いつでも反応を返せるよう準備がされていること”が正しい。Macintoshのハイパーカードなんてのがそれを目指したものだ。あれは楽しい。なのにどういうことか，マウスカーソルはどこにでも動くのに，クリックの効く位置が限られているソフトが多いということである。あるときにはある範囲に置かれたアイコンしか，別のときには別の範囲に置かれたアイコンしか反応しないというのは極悪非道なのだ。

もし，そういうグラフィックツールがあったら，誰もが極悪非道だというだろう。画面にペン先選択の表が出ているのに，別のメニューでペン先選択モードにしなければその表が使えないとなったら，誰もがそのソフトを非難するはずだ。

どうやらこういう習慣があるらしい。そのときに使えないメニューやアイコンに関しては，“網をかけたり，薄い色で表示”してそれを示す。ダンジョン・マスターやCANVAS PRO-68Kなどではそうしている。こういう習慣をうまく使うべきなのだ。

ぐちゃぐちゃ能書き垂れているのも芸がないので，例を挙げる。

3Dダンジョン編

3Dダンジョンの移動だ。ダンジョン・マスターはみんなお馴染み。移動は左下の移動アイコン群で行い，視界はものを投げたり，取ったり，叩いたりするのに使う。そういうことになっている。が，泉大介氏曰く，「ねじ式のほうがいい」。

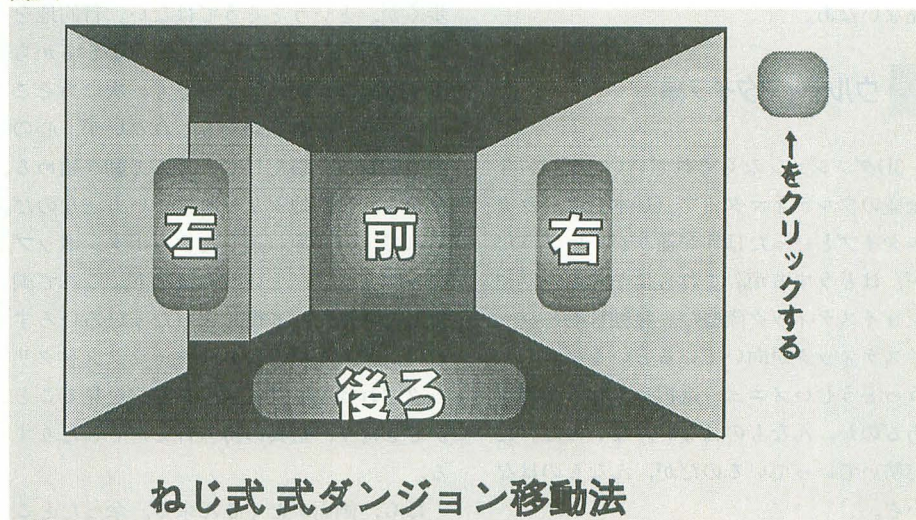
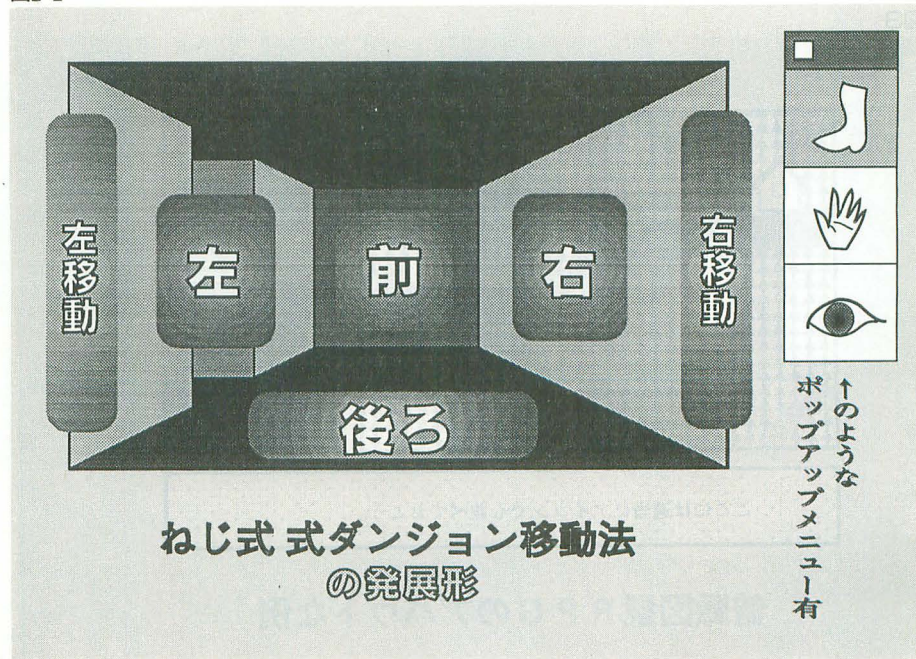
ねじ式のダンジョンモードは図2-1のようなシステムになっている。ダンジョンのグラフィックをクリックした方向に進むのだ。ねじ式のダンジョン自体は非常にわかりにくいものだったが，この移動法は確かにわかりやすい。

私は考えた。(泉大介氏の意見を大幅に採用して)この方式を応用して，ダンジョン・マスターのようなシステムにも使えないだろうか。そこでできたのが，図2-2だ。まず，右へ動いたり後退したりという動作を枠の外へ置き，中は実に感覚的な操作でいけるようにした。こうすると，マウスの移動量が多くて面倒，という声もあろうかと思うので，それぞれのエリアはもう少し隣接させてもいい。ここで，マウスの移動量が少ないものを取るか，わかりやすいほうを取るか，だな。人間の集中できる視界というのはそれほど広いものではないので，ダンジョン・マスターのように視界ウィン

図2-1

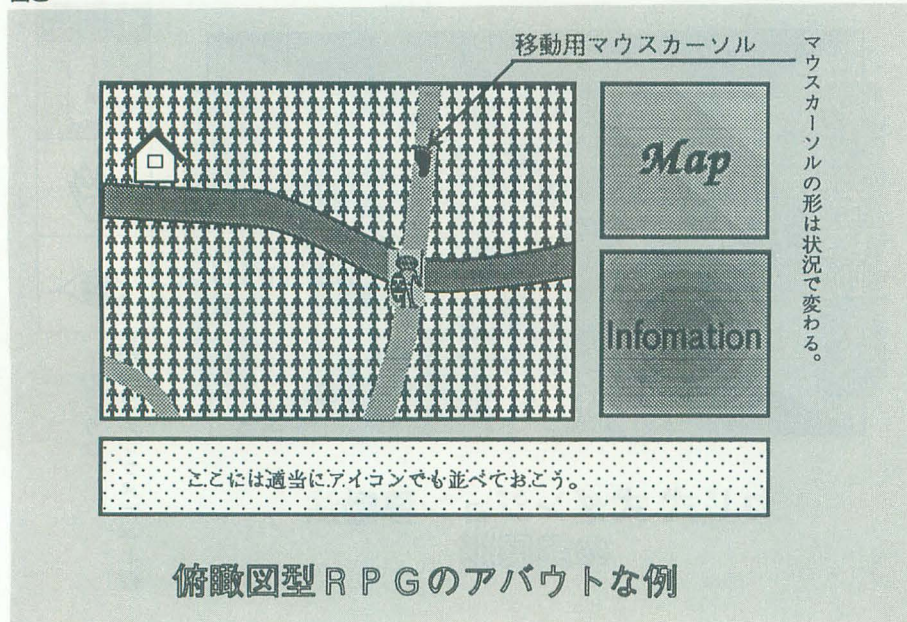
図2-2

ドウを凝視する必要があるゲームでは，多少面倒でもそこから目が離れないようなシステムのほうが親切だ。

で，これでは移動はできても，投げたり拾ったりの動作はできない。そこで導入するのが，ポップアップメニューだ。右ボタンを押すと，“足”と“手”と“目”のアイコンがポップアップする。足は移動，手は投げたり拾ったり叩いたり，目は見たり調べたりである。もちろん，選んだアイコンがそのままマウスカーソルになる。鍵を使ったりするときは，ダンジョン・マスターと同じで，鍵のマウスカーソルになる。どうだろうか。

そういえば，ポップアップメニューを“真剣に”使ったゲームっていうのはなかなか見ないなあ。

ウルティマタイプ編

3Dダンジョンならそれでいい。では，今全盛のウルティマタイプ（日本ではドラクエタイプといったほうが通りがいいようだが）はどうするか。これらはキーボードやジョイスティック隆盛だ。移動行為にジョイスティックが向いているというだけで，うっとうしいメニュー選択をさせる必要があるのか。んなものはない。ないのだ。ただ勢いでいっているのだが，んなものはないぞ。

というわけで，図3である。要は，移動をなんとかすればいいのだ。

で，私はこうする。図を見よ。画面には15×15マス程度の俯瞰図が表示される。これはなんでも同じだ。このマスの中では，上の例にならって，マウスカーソルが靴の形になる。プレイヤーはこのマップに表示されている中で，「直線で行ってかまわない程度の目的地をクリックする」だけでいいのだ。すると，キャラクターはそこに向かってマップをスクロールさせながら歩いていく。ただこれだけ。途中でなにかイベントがあったら止まるようにしておけばいいだけだ。うっとうしければ，マスクしたイベント（一度しか用のない石碑とか）に限って無視できるようにすればいい。

簡単だ。我々が移動するとき，一歩一歩歩くか，というとそうではない。目的地を決め，「あそこまで行くぞ」と決心してから動くではないか。ゲームでも，次はあそこへ行こう，と無意識かもしれないが，心の中で密かに一瞬だけでも考えて動き始める。だから，目的地クリックはいい方法なのだ。それで，目的地に達したら右ボタンポップアップメニューというお決まり。これで調査したり休んだり街に入ったりいろいろすれば，楽ちんだ。自分のキャラクタをクリックすればインフォメーションを見ることができたり，道具の持ち替えができたりする。

ほら，画面がたくさん余る。余ったとこは全体マップなり，いろいろ特殊なアイコンなりを並べたり，メッセージエリアにするのもいい。戦い用アイコンくらいはダンジョン・マスターのように別に用意したほうがいいだろうな。

というわけで，ジョイスティックをセットするだけのスペースもなく，キーボード操作の面倒さ（ウルティマ！）やカーソル移動による面倒なメニュー操作から解放されるのだ（実際に作ってみて，やりにくかったらどうしよう。ちょっと不安）。

図3

俯瞰図型ＲＰＧのアバウトな例

アドベンチャーゲーム編

変なことを考えたついでに，アドベンチャーゲームにも手を伸ばしてやろう。だいたい，アドベンチャーゲームというのは独自の世界を勝手に構築してしまったためにユーザーインタフェイスも何もあったものではない。アドベンチャーゲームという名の特殊な法則が支配する世界があって，その世界ではどんなことでも許されてしまうのだ。人間がソフトの都合に合わせてしまう好例といえよう（そこまでいってしまうと，RPGだってたいていそうなのだが）。

私としては，オブジェクト嗜好っぽいものにしたい（補足：最近の俗世では，先に対象を指定してから動作を指定することをオブジェクト指向というらしい。ここではオブジェクト指向言語と区別するために，オブジェクト嗜好とでもしておこう）。

だいたい，いくら絵がきれいでも，絵と動作がまったく別物として管理されるのはどうも好きになれない。だいたい，見る，とか調べる，なんていうコマンドがよくない。よし，見るぞ，なんて思ってから見るやつなんていない。これはアメリカ製テキストアドベンチャーの名残をただひきずっているだけである。21世紀を前にしてすることではない。

そういうわけで，オブジェクト嗜好である。図4のような画面を考えてみた。

とりあえず，窓とメモボックスとアイコンテーブルからなる。窓っていうのは，X68000内に構築されたアドベンチャーワールドを覗き見るための窓である。ゲームはすべてこの中で進められる。メモボックスというのは，プレイヤーが自由に使えるメモ帳である。スクロールバーを使ってスクロールができるので，長い文章も打ち込

める。字がでかいといまひとつみっともないので（おそらく512×512ドットモードだろうから），12ドットのフォントがいい。アイコンテーブルはおまけ。ここにはデータのセーブやら，窓内の情報の切り替えやら，自分の状態（持ちものなど）のチェックやらを行う。

私が考えたのは，移動アイコンをクリックすると窓内のグラフィックが地図になり，行きたいところを選んだり，その場所の情報を得られたりし，持ちものアイコンをクリックすると，窓内に鞄の中身をぶちまけられたり，電話アイコンをクリックすると，事務所（リバーヒルものの探偵アドベンチャーを想定）と連絡がとれたりするのだ。

肝心の窓である。ここは一見，普通のアドベンチャーゲーム風である。適当に見たり調べたり取ったりしたいものを左ボタンでクリックするとそれがズームアップする。さらに細かく調べられるものはもう1段階ズームアップする。手紙などは，それで内容を読めるようにする。右ボタンはポップアップメニューである。ここには，ズームダウンやゲットなどがあり，細かい動作の指定が行える。人物と会話をするときもポップアップメニューを使う。

まず排除したいのが，いま主流の，主人公の独り言である。プレイヤーが主人公なのに，ゲーム内主人公がペラペラしゃべるのはうっとうしいことこの上ない。「ここはなんとかだよな」なんてことをいかにいわせずにすますか，がひとつの問題である。どうしても必要なときは吹き出しを出す。登場人物が喋るときも吹き出しを出す。

つまり，「見る」とか「調べる」という行為は画面を凝視することによって達成され，プレイヤーは画面情報からのみ情報を得る。決して自分の独り言ではないのである。X68000クラスのマシンなら簡単にできるはずだ。

結局は2つにひとつだよん

いろいろとユーザーインタフェイスデザインは多くの可能性とともに混乱を伴う。どういうものを選ぶかは2つにひとつだ。ユーザーを慣れさせてしまうか，ユーザーにこびるかである。

ユーザーにこびるというのは，ファミコンのように単純なユーザーインタフェイスを与えて慣れさせるか，ATMのタッチパネルのように画面メッセージとユーザーの動作をできるだけつなげるかである。

慣れさせるというのは，マウスやキーボードのオペレーションである。ある程度複雑な動作を指定しようとすると，こびるだけでは限界がある。ユーザーに簡単だよ，簡単だよといって騙しながら慣れさせてしまうという最近のパソコン業界やビジネス業界の常套手段もあるが，あれはサギだ。

図4

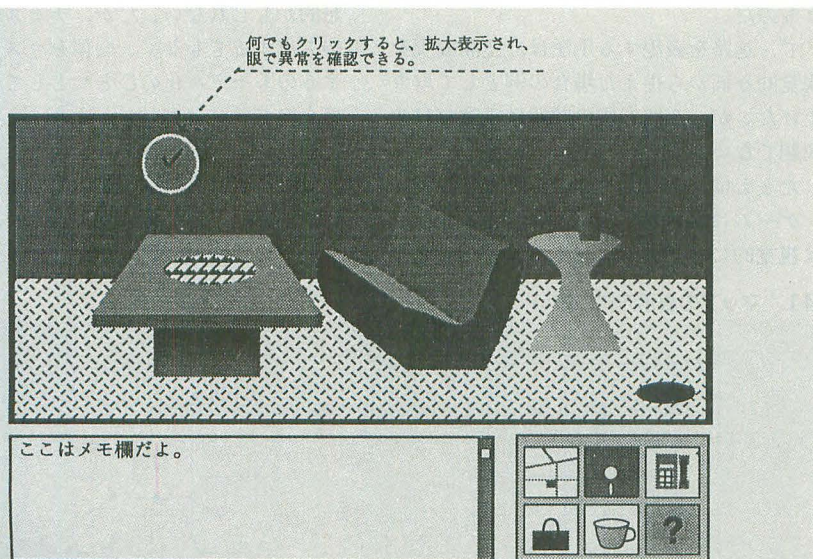
じゃあ，慣れやすいものを作ってやろう，慣れたらあとは快適だと思わせる環境を作ってやろう，というのが正しい，と思う。Macintoshにはそれを感じる。Macintoshはユーザーインタフェイスガイドを作り，こういうユーザーインタフェイスを基本にしてアプリケーションを作ってください，とまでやっている。

マウスは非常に単純なオペレーションで，人間の行う動作とけっこう対応している。ドラッグはつかんで引きずりまわすことや指で囲って範囲を示すことだし，プルダウンメニューはトイレットペーパーを引き出す感覚だし，ダブルクリックは尻ごみしているやつを後ろから突き飛ばす動作だし，クリックは指をさすことだ。これだけならワンボタンマウスである。2ボタンマウスの右クリックはちとややこしいが，ポップアップメニューはお尻のポケットからメモを取り出す感覚だ。キャンセルというのはマウスに“指を離してチャイする”という機能を持たせられない代わりだ。

だから，Macintoshはわかりやすいといわれるのである。マウスを使ったユーザーインタフェイスは便利さを追求するあまり，状況に応じて右ボタンと左ボタンの機能が変わってしまうような複雑さをもってはいけないのだ。それはユーザーに“特別な記憶”を要求するものであり，慣れれば快適だというのは通用しない。それを通用させてきたから現在のような怪奇な状況ができあがり，新規参入ユーザーに勉強を強いる元になったのだ。

現実からあまり遊離しないユーザーインタフェイスはパソコンの中に繰り広げられる仮想世界（時代はバーチャル！）に没入させやすい。これをフレンドリーという。こういうフレンドリーさと，ユーザーの動作がどういう影響を与えたかというフィードバックが見えやすい設計。せっかくマウスを使い，グラフィックベースの資源をパクパク食べる環境を使うのなら，ここまで考えてほしいのだ。

＊

そういうわけで，どこがゲーム特集かわからなくなった原稿は終わる。ちょっと散漫な気もするが，なあに，そういうときは“ちょっと散漫な気がする”と自分で先にいってしまえばいいのだ。へへへ。

「わかる」ためのしくみを考えよう

ゲーム空間へのインタフェイス

Saitou Susumu 斎藤 晋

ゲームを楽しむことは，ゲームの中で考え，行動するということです。でも，プレイヤーがゲームの持つ空間を自然に感じるためにはどんな工夫が必要でしょうか。ここではマップを見せたり操作させたりする際の問題点を見ていきましょう。

ゲームを立ち上げ，スタートさせるとそこにはそのゲーム独自の世界が広がっている。プレイするにつれ，その空間のとりこになってしまうことも多い。

さて，これまでも私はゲーム内における迷路やマップなどの空間的な問題を扱ってきたが，今回はこれらをユーザーインタフェイスの問題と絡めて考えてみることにしたい。

いうまでもないが，パソコンのディスプレイは基本的に平面だ。そのことを私たちはよく知っている。ビジネスソフトには立体的な３Ｄグラフを作成するものがあるが，図法的な工夫はいってみれば約束ごとである。３つの座標軸にそって描かれたものは立体を表しており，ルールに従ってグラフの形を理解しなさいということだ。べつに空間的リアリティは必要ない。電信柱ならいざしらず，グラフの影からはお笑い芸人を登場させる必要はないからだ。

ところが，ゲームには視覚，触覚，そして想像力を動員させて，ゲームの持つ世界観をプレイヤーに与えなくてはならない。ゲームが必要としているわかりやすさはビジネスソフトとはレベルの違うものであるということだろう。

ゲーム空間のモデル表現

ゲームはその性質によってさまざまな空間表現の手法を採用している。空間表現というといまひとつわかりにくいので，具体例を挙げてみよう。

まずはお馴染みのダンジョンの表示だ。３Ｄ迷路だったり，天井をとっぱらって上から眺めるトップビュータイプだったり，サイドビューの断面図だったりする。なんだ，絵の描き方の問題か，と思われるかもしれないが，これらの手法の違いはプレイヤーと空間との関わり方に大きく影響する。視点の違いは空間の迷宮性すら変えてしまうものだ。

で，迷路を表現する手法は，空間表現を視覚的な面から捉えた場合の例として取り上げた。が，空間表現の問題は視覚だけの問題でない。

たとえば，大戦略などのシミュレーションゲームにおけるヘックス表示。これなどは視覚的にもゲームに対するイメージを与えるものだが，もっといえばゲーム空間の特性をも決める重要な要素だ。

ウォーシミュレーションがヘックス表示を採用するのはボードゲームからの伝統だが，この方式は隣接するマス目が最も多くかつ移動ステップと移動距離とが一致するように考案されたものだ。通常のセクションスケール（方眼）では隣接するマス目が４つしかない。つまり90度ごとの４方向にしか移動できないということだ。45度方向の目的地に進む場合は直線距離より40パーセントもの回り道を進まなければならない斜めに進めるとしたら８方向だが，それでは逆に移動距離が40パーセント以上も大きくなってしまう。ヘックスなら60度ごとの６方向に隣接したマス目があり，最も不利な30度ずれた方向でも15パーセント程度のロスですむ。

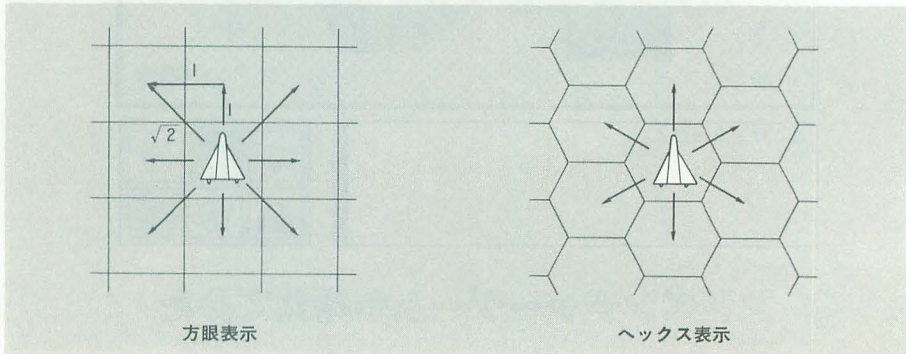
もちろん，マップ全体がマス目を行動の単位とするシステムに異を唱える人もいるだろう。マス目単位ではなく任意の方向に任意の距離だけ進めるというシステムが理想的かもしれない。だが，チェスや将棋を持ち出すまでもなく，全体をマス目に分割するのもモデル化のひとつとして有効だと思うのでそれはよしとしよう。

とはいえ，ヘックス表示はあまり歓迎できたものではない。空間のもつ面白さを強調するというのならヘックスでも，クォータービューでもよいのだが，より正確かつ

「SUPER大戦略」ではフルマップやローカルエリアマップなどのウィンドウ配置を自由に変えられる。なお，メニューはプルダウンだが，WP.Xなどと違ってクリックすると，何かを選択するまで開いたままになるタイプだ。

図１　マップの表示方式の違い

方眼表示

ヘックス表示

自然に空間を把握するという視点で考えるならヘックスには疑問をもたざるを得ない。すくなくとも近代社会で教育を受けた人間が空間を測る基準は縦・横・奥行きという3次元の座標軸だ。平面にすれば，広さや領域を考える単位は，なんといっても矩形である。いずれにしても直交座標系でないと気持ちが悪いのだ。

問題解決への道

マス目にはやはり碁盤の目が望ましい。としたら？　先ほどの移動方向と距離の問題はどうすればよいか。実はそれほどたいした方法は必要ない。ざっと次のような手が考えられるだろう。

1)　移動可能な距離に基づき，近似的に移動可能なマス目を定める。

2)　マス目の解像度を高くし，相対的に移動可能な距離を大きく設定する。

1)は要するに，図2-1のようなことだ。確かにボードゲームではプレイヤーが移動可能なマス目をボード上で数えなくてはならないから面倒だ。途中に山などがあって移動距離に影響する場合などはさすがに計算などしてはいられない。が，ここで話をしているのは高い演算能力をもつパソコンだ。移動可能なエリアを演算して表示するくらいわけはない。

問題は，移動力（移動可能な距離）が小さいユニットがある場合だ。たとえば，移動力が1の場合はそのまま最初の問題に突き当たる。そこで，2)のような方法が必要となる。ボードでは実際にミニチュアのユニットをマス目の上におかなければならないから大変だ。解像度を上げるということは，ボード全体を大きくするか，さもなくば，マス目だけでなくユニットも小さくしなくてはならない。

これもパソコンの場合は解決策がいくらでもある。マップを大きくすることはメモリの使い方次第でいくらでも可能であり，部屋が狭くなる心配はない。問題はむしろ表示方法だ。また，画面モードを解像度の高いものにすれば（たとえば，786×512ドット），画面上で多少マス目が小さくなっても情報量はそれほど失われない。ユニットの表示も小さくなるが，アイコンとしての機能が損なわれない範囲であればあとはソフトの力でいろいろと手が打てる。

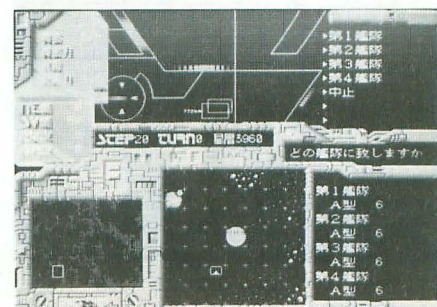
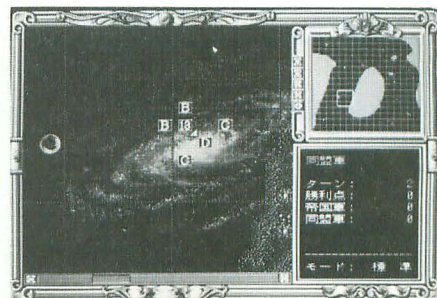
というわけで，関連した具体的なユーザーインタフェイスを考えてみよう。

マップと空間

多くのゲームではかなり広いエリアをゲーム空間として持っている。RPGなどではその全体が迷宮のようなものとの解釈もでき，ゲーム中に全体を見渡す必要は必ずしもないが，シミュレーション的なゲームでは戦略的にいって全体図を掌握する必要がある。先に登場した大戦略などでは，マップ全体に注意を払いながら，局所的なエリアで行動を起こさなくてはならない。

SUPER大戦略68Kではオペレーションの対象となるローカルエリアマップ（ビューマップ）と全体の地形を見渡せるフルマップとがあり，標準状態でゲームを開始すると，ビューマップが大きく，フルマップを右上に小さく表示する。が，これらはプルダウンメニューからレイアウトの変更を選ぶことによって逆にしたり，ユーザー定義機能によってさまざまに変えることができる。これはなかなか親切そうだ。

ところが，実際の使い勝手は見掛けとは違う。問題はフルマップとビューマップの2つのマップをいかに有機的に機能させるかということだ。特にユニットの1ターンあたりの行動範囲が，ビューマップに対して上回るような場合はなおさらのことだ。

フルマップ上のどこかのポイントをマウスで指示すれば，そのエリアがビューマップに表示される。これらのマップ表示を行うほとんどのゲームでも同様だ。異なるのはビューマップのオペレーションだ。

全体図とビューマップの配置例。（上）が「銀河英雄伝説II」，（下）が「シュヴァルツシルト」だ。

SUPER大戦略68Kではどうなっているか。ビューマップのスクロールにはマウスカーソルをビューマップの枠のそばに持っていくと，カーソルがスクロール方向を示す矢印に変わる。ここで左クリックすると画面がスクロールするというわけだ。これは，ビューマップのはじっこでは，左クリックによる通常の操作と間違えやすい。カーソルの形を確認しながらプレイする人はいるだろうか。狙ったところに素早く移動して左クリック，これがポインティングデバイスとしてのマウスの醍醐味なのである。

この点，シムシティーでは機種によっても操作が違う。作法の厳しいMacintoshでは，専用のスクロールボックスとスクロールバーを使う。IBM-PCやFMTOWNSでは画面の端にマウスカーソルがぶつかるとその方向にスクロールする。そして，X68000とPC-9801では，ビューマップウィンドウの枠の外で右クリックである。

図2-1　1)の方法による移動力の設定例

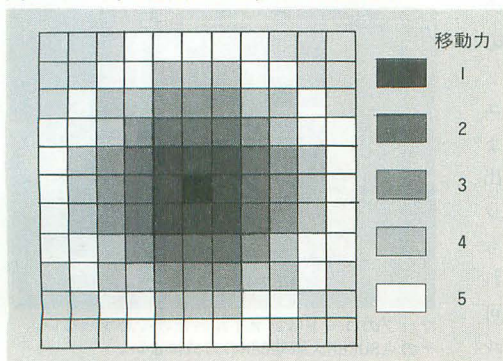
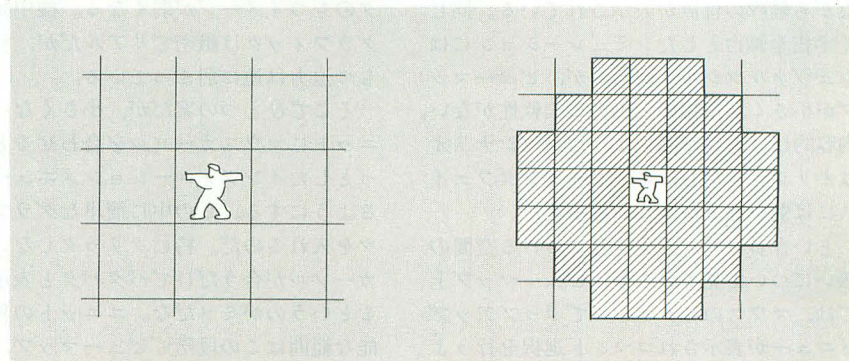
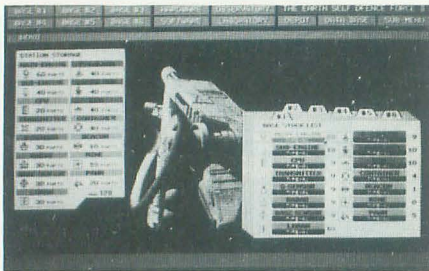
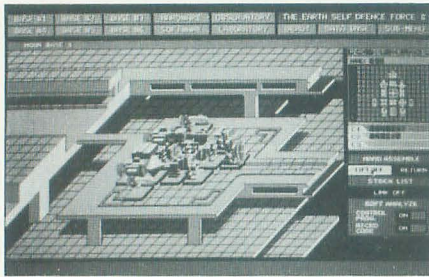



図2-2　2)の方式による移動力の設定例

ユニークなデザインの「FAR SIDE MOON」。(上)は戦闘機の部品をアセンブルしたところだ。補給部品の選択も凝っている(下)。

Macintoshの場合はワンボタンでもあるし，他のアプリケーションとの操作性の統一という明快な大義がある。が，他の方式に比べて効率はややマイナスだ。IBM-PCやFMTOWNSの方式はダイナミックで気持ちいいが，マウスの移動量が大きいのと不意に余計なスクロールをしてしまうことが多い。その不意のスクロールを避けるのがX68000とPC-9801の方式だが，これもマウスの移動量が大きいのと，ウィンドウ内では右クリックが整地機能に割り当てられているから誤って大事な施設を壊す心配もある。というわけで，いずれも一長一短。必ずしもどれがよいとはいえないが，一応納得のいく線ではある。

空間へのアクセス

これらのマップの扱いで，ビューマップとフルマップとの関係がうまく連動していると思ったのは銀河英雄伝説IIである。メインとなるビューマップは固定だが，大きくて美しい。また，フルマップにも小さいながら艦隊の位置が表示されている。同じく宇宙を舞台としたシミュレーションにはシュヴァルツシルトがあるが，ビューマップが小さく，情報の入出力に柔軟性がない。内容的には人気が高いようだが（シナリオはオリジナルだし）ユーザーインタフェイスには弱い部分があるのも事実だ。

というわけで，銀英伝IIにおける空間の扱いについて見ていこう。ビューマップ上では，マウスの左クリックでポップアップメニューが表示されコマンド選択を行うようになっている。惑星や艦隊の上でクリックすれば必要な情報を得たりコマンドを与えることができる。また，なにもないところでクリックして惑星というのを選択する惑星の一覧とそれぞれの属性を示すウィンドウが開き，マウスカーソルで惑星名をなぞると次々と表示が反転するだけでなく，全体を表すフルマップ上で惑星の位置を教えてくれる。これで各惑星の位置関係が素早く把握できてしまう。

先のSUPER大戦略68Kでもユニットの一覧とかがあるが，ユニットを選択してメニューを閉じると初めてマップ上の位置を教えてくれる。なんでもないことのようだが，ゲームのわかりやすさというのはこのへんで差がついてしまうものなのだ。

さて，この銀英伝IIだが，ビューマップのスクロールがまたいい。基本的には右クリックでスクロールするのだが，スクロールの方向はカーソルがビューマップ内のどの方向に近いかで決まる。しかも，右クリックしたままでマウスを動かすことができるため，自由自在にスクロール方向をコントロールできるのだ。おまけにこれがメインの座標系とバックの銀河の多重スクロールときた。レスポンスもいい。銀河の奥行きが感じられる。

シミュレーションゲームにはプログラミング技術に不満の残るものが多いけど，この銀英伝IIは水準が高いようだ。なお，スクロールは右クリックだけでなく，ビューマップのまわりにはスクロールバーや4方向のスクロールボックスもついている。いたれりつくせりといったところか。

情報に奥行きを持たせる

先ほど，2)のマップの解像度を高くする件で，マス目やユニットの表示が小さくなっても，ソフトの力でなんとでもなると書いたが，そのへんにも触れておこう。

ユニットが小さくなるということは，一覧性が高くなることと引き換えにキャラクタのもつイメージが弱くなる。機甲師団のグラフィックは緻密でリアルだが，戦車のもつ迫力は逆に弱まっている。

そこでひとつの案だが，小さくなったユニットにマウスカーソルを合わせるとちょっとしたインフォメーションメニューが出るようにする。この中に簡単なグラフィックを入れるのだ。特にクリックしないでもカーソルが合うだけでパタパタと表示されるというのがミソだな。ユニットの移動可能な範囲はこの段階でビューマップ，フルマップの双方に表示するようにする。もちろん，右クリックは銀英伝IIの方式でマップのスクロール用に使いたい。で，左クリックでユニットが反転して選択され，そのままドラッグすればアイコンのごとく持ち運べる。これがすなわち移動である。また，ユニットが選択された状態で右クリックするとポップアップメニューで戦闘，補給といった軍事コマンドが実行できるのだ。

いろいろ言ったけど

ちょっとした問題点（ヘックスを否定すること）から出発したインタフェイスの提案は連鎖的にさまざまな問題を解決しなければならないことに気づかせてくれる。しかし，ここに紹介したアイデアは私が考えたわけでもなんでもない。よく出来たソフトが同様の問題をどう扱っているかを調べたまでのことだ。ただし，より細かい話に入っていくと全体との統一性やバランスが崩れるものもあるので注意が必要だろう。

さてと，ユーザーインタフェイスなどという大上段に構えた言葉につられて，つい正論をふりかざすような書き方をしてしまったが，こういった枠組みの中では語りにくいことも多い。デザインがよくて理屈はとりあえず置いておこうという場合もある。

たとえば，FAR SIDE MOON。レスポンスが悪く，ただのプルダウンメニューにディスクアクセスが入るというルール違反や，ディスクコピーはドライブ1から0へ（なのに0ドライブにゲームディスクが入っていないと暴走する）という明らかな誤りもある。しかし，このゲーム画面を見ると日本人離れしたデザインについ気を許してしまうのだ。

最後に，今回槍玉に挙がってしまった感のあるSUPER大戦略68Kについてフォローしておこう。マップのロードにイメージ情報があるのはとても評価できる。どうしてほかのソフトはやってくれないんだろう。というわけで，今回はここまで。

マップのロードはファイル名だけでわからない。その点SUPER大戦略68Kの方式はよい。

インタフェイスを見直して

アドベンチャーゲームを救え

Nishikawa Zenji 西川 善司

最近は話題作が少なくなったアドベンチャーゲームですが，システムやインタフェイスを検討すればまだまだ魅力的なデザインが可能なジャンルのはずです。アドベンチャーの歩んできた道程を振り返りながら，問題点を探ってみましょう。

昔のパソコン小僧の脳味噌には以下の等式がインプットされていた。

ゲームセンター＝アクションゲーム
パソコンゲーム＝アドベンチャー，RPG

だから，パソコンを持っているゲーム小僧は，小銭で膨れた財布をポケットに，腕まくりをしてゲームセンターに入っていったし，万が一パソコンにアクション系のゲームが出たとしても，「パソコン上でのアクションゲーム」という多少なりとも差別的な目で見ていたのだった。

しかし，アドベンチャー（以下ADVG）はパソコンならではのものであった。パソコンを持っていないゲーム小僧はゲームセンターのゲームとはひと味違ったこのパソコンならではのADVGに対して，まるで異国の民俗衣装を着込んだ美女を見るような，憧れの眼差しを向けていたものであった。私の家には毎日のように友人がMZ-700の「不思議の森アドベンチャー」をやりに来たものだった（この友人は私が解く前に解きやがった，畜生，ゆるせねえ，はあはあ）。

ところが，最近では上のようなことが一概にはいえなくなってきた。パソコン側の頭痛のするようなハードの進歩のおかげもあってパソコンでアクションゲームも楽しめるようになってきためだ。また，家庭用ゲーム機の異常なまでの普及によってゲームセンターにまで足を運ばなくてもアクションゲームは遊べるという状況にもなっている。

とにかく，いろいろなゲームが至る場所で楽しめるようになってきている今日でも，ADVGはやはりパソコンならではのものである。が，しかし，どうも最近は廃れてきているような感じが……。

ソフトハウスVSユーザー

どうも最近は心に残るADVGが少ない。去年1990年を振り返っても思いつくのは「闇の血族」（システムサコム）ぐらいか。システムサコムには申し訳ないがこれは決して「闇の血族」が1990年を代表するような名作だったため，だからではない（と思う）。最近は発売されるADVGの絶対量が少ないのだ。

どうも日本のソフト業界は，

RPG＝売れる
ADVG＝売れない

と思い込んでいるためらしい。ADVGが売れにくい理由はひとつ，需要側が，

ソフトの値段が高い割にはすぐ終わると思い込んでいるためだ。その気持ちは分かる。たとえば去年PC-9801用に発売された「サイレントメビウス」は14,800円もするくせにせいぜい5，6時間で終わってしまう。ゲームセンターに14,800円持って行ったら1ゲーム50円で5分遊べるとしたら24時間以上遊べる計算になるし，秋葉原やヨドバシなんとかとかへ行けばメガドライブ本体だって買えてしまう（それでも売れたっていうからオタクパワーには敬服するぜ）。

そしてソフトハウス側は，先にいったようなことを思い込んでいるうえに，RPGなどと同じぐらいの手間がかかるのに（売れないから）割が合わないといい張るため，ADVGは廃れる一方。

これはまずい。ヤバイ。

私はどちらかというとユーザーを代表する立場にいるわけだが，まあ，いわせてもらえばADVGの売れなくなった原因はソフトハウスにあると思う。

ここ数年のADVGをみるとシナリオやグラフィックの進歩は誰も否定しないところだが，一番重要であるユーザーインタフェイスは停滞気味だ。まずはこのあたりに切り込んでみたいと思う。

コマンド選択の台頭

ADVG界のピューリタン革命といわれる「コマンド選択」。それまでのADVGでは動詞，名詞を入力しなければならず，だれもが言葉探しと「ソレハデキマセン」攻撃に悩まされていた。私もX1の「サザンクロス」（バンダイ）で宇宙船の「コンソールパネル」を「ツカウ」と「トオクテテガトドキマセン」といわれ悩んだことがあった。「コンソール　イク」「コンソール　ヨル」……思いつくまま入力するがすべて「ソレハデキマセン」。頭にきて「バカ」と入力すれば「ソレハアナタデス」……ウッキー！結局，ある日ふと思い浮かんだ「コンソール　チカヅク」が正解だったわけだが，先に進めた嬉しさと同じくらい開発者への怒りのほうも大きかったものだ。

そもそもADVGの世界では開発者は「神」であるからその「神」の意図に反する行動はできなくてしかるべしである，という考えもなくはない。しかし半面，シナリオの制作者としてはプレイヤーの全員に最後まで見てもらいたいという願望も強いわけで，まあ，そういった事情から，とれる行動を先に挙げておき，それらからユーザーに選んでもらう方式，「コマンド選択方式」が台頭してきたのである。

ところがこの「コマンド選択」には副作用もあった。それは必然的にゲームのプレイ時間が短くなるというものだ。なにしろ，プレイヤーが無駄なコマンド入力をするこ

とがまったくないのだからシナリオはバンバンと先へ進む。これを憂えたソフトハウス側は「コマンド選択」方式でもプレイ時間を長くする方法を考えてきた。結局それは「言葉捜し」と同様の苦痛をプレイヤーに与えるということに気づかずに……。

不要の選択肢

ソフトハウス側はシナリオ進行上関係ない選択肢をメニューに忍ばせるようになってきた。そういったコマンドを選択すると「ソレハデキマセン」と同じようなユーザーを小馬鹿にしたようなメッセージを返す。最近はこの傾向が「見る」というコマンドに著しい。「見る」を選ぶとその場面にある物体の名前がズラーッ。えぇーい，私は主人公の持っている服のブランドには興味はないのだ，テーブルがいくつあるなんてのは絵を見りゃわかる！　……ふぅ。また，「話す」にも腹立たしいものがある。たとえば画面に人がいないのに「話す」と「話す人がいません」……。くーっ，絵を見りゃわ・か・る・よ！　それに「取る」も……。「そんなものは取れません」「重たくて持てません」……うっきーうききっきー。とゲーム中はパーマン２号に変身してしまいそうなときがしばしば私にはあるのだ。

しかし，ゲーム進行上に関係ない遊び心のあるメッセージを入れたい開発者の気持ちもわかる。だったら一度使用した名詞やコマンドは文字の色を変えるとか，消えるとかしてほしい。また，「見る」と「調べる」はひとつにまとめるべきであろう。１，２回「見る」と新たな物がみつかりさらに「調べる」「取る」というパターンはうんざりだ。「コマンド入力」時代でも「見る」と「調べる」は同義語だったぞ。

メッセージはADVGの顔

さて，ADVGといえば文章，メッセージは重要なファクターである。どうも「コマンド入力」時代から日本のADVGは絵ばかりを重要視する傾向がある。しかし，考えてもらいたい。文章は絵に優るとも劣らぬくらいプレイ中に見られているのだ。

まず禁則処理（。，」などが行の先頭に来ないようにすること）にもうちょっと気配りが欲しい。これはゲーム本編には関係ないことではあるが浦川氏の記事にもあるような「ゲームの質感」に関わってくると思う。たとえば感動的なシーンの締めゼリフの最終行が，

」

だったらどうか。私なら狼男が月を見たような呻き声を上げ開発者に呪いの文句を捧げるところだ。この文字の「禁則処理」はプログラムの技術的な問題ではないので要は作り手の心構えの問題と思うのだ。

また，日本人は横文字アルファベットが大好きなくせに英語の禁則処理に対してはてんで鈍感だ。オープニングやエンディングには英文をロールしたりするのを見掛けるが，

One little,two little,three little ind
ians

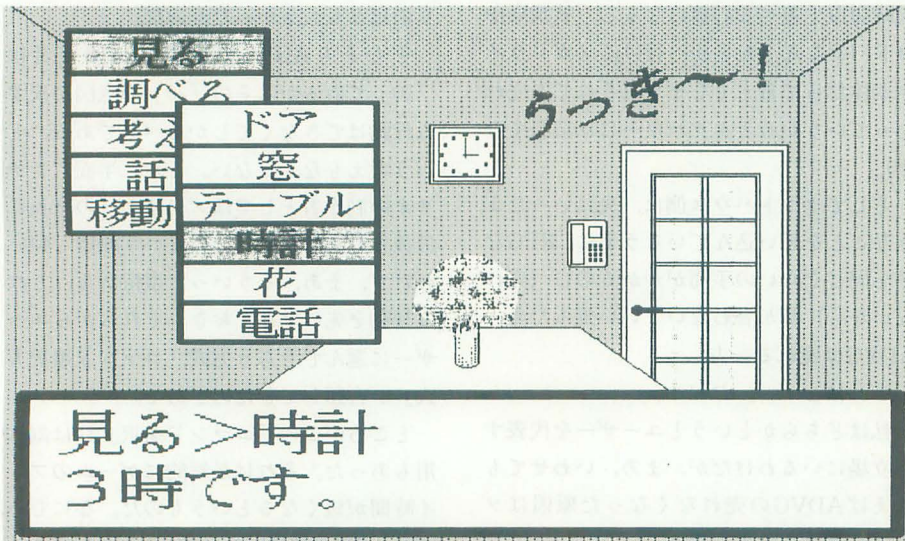
「syllables」（音節）を完全に無視した改行を平気で行っている。しかしこんな基本的なことは中学校で習ったと思うんだけどねぇ。

さて，最近は少なくなってきたが以前はメッセージの出し方に問題のあるゲームが多くあったものだ。たとえば「JESUS」（エニックス）。１文字１文字ポチポチとゆっくり出力しプレイしててやたら疲れた記憶が私にはある。あれをプレイしてから１週間は歩き方や呼吸があのポチポチと同じリズムになってしまったほどだ。まあ，最近は「CAPS」をロックしていると高速に出力するものやシステムサコムの「ノヴェルウェア」シリーズなどでは可変のものなど工夫がなされるようになってきた。

が，最近のADVGはシナリオが凝ってきたうえに文学的で難解な表現があったりして，ゲーム画面の表示エリアに映っている文章だけでは理解しにくいときがある。推理小説を呼んでいても数ページ戻って読むことがあると同じにADVGにも文章をスクロールバックできるような機能がぜひ欲しい。どこかのソフトハウスが以前出力したグラフィックが見られるというのをなんとかシステムと，たいそうな名前をつけて呼んでいたが今度は文章のほうにもそういった機能を付けてほしい。

表示している文章を少々スクロールバックすることは技術的に難しいことでないし，TOWNSやX68000のような大メモリのマシンなら表示した文章をそのままワークエリアに待避しておきそれをプレイバックするだけでもかなり前のものも読むことができるはずだ。

アイテムの出方とハマリ

ADVGには謎解きの鍵となるアイテムがいろいろ登場するものである。「コマンド選択」方式がADVGの主流となってからゲームが進行しない状態，いわゆる「ハマリ」に陥ったときには少なからずアイテムが関係しているものだ。そのアイテムを手に入れるためにはたいていのADVGはまず，「見る」そして「調べる」最後に「取る」のプロセスを取るわけだが，ゲームの進展状況によってはいくら「見る」しても「調べる」してもアイテムが見つからないことがある。プレイヤー側はプログラムのフラ

グがどうのこうのといったことはほとんど気にしていないから「あそこは以前にあんなに調べたのに何も見つからなかった」と思い込み，結局「ハマリ」に陥るのである。

ところでADVGの「ハマリ」はあってはならないものなのか。いや，そうともいいきれない。ある状況において誤った行動をとったために失敗する，ゲームオーバーとなるというのはゲームと名前がつくからには必要なものだとは思う。しかしプレイヤーにその「ハマリ」をわからせるというのは大事なことだと思うのだ。プレイヤーにプログラム的な，デジタル的なことを感じさせてしまうADVGは「出来損ない」といってもいい。

「サイオブレード」（T＆Eソフト）では捕らえられた主人公が牢屋の鉄柵ごしに外を観察するシーンがある。ここで外に止まっている車のナンバーを覚えていないと先のシーンでコンピュータで車の行き先を調べるときに「ハマリ」に陥りゲームオーバーとなる。これはよい「ハマリ」だろう。さらに「どこかに見落としていたことがあるはずだ」のご通達。これはなかなかうまい方法だ。むやみやたらに歩き回らされ永遠に進展しないゲームをさせられるよりもずっと気持ちがいい。

さて，話を少し戻して，では「アイテム」による「ハマリ」はどうしたらいいのか。これには解決策がなくはない。たとえば，ある状況では「取る」ことができなくても「調べる」ことによって見つけられるようにしておけばよいのだ。たとえば，机の上に「ペン」があり，これを「取る」すると「こんなものは必要ない」といったメッセージを初めは返しておき，状況の進展により必要に迫られた時に取れるようにすればよいのだ。これならシナリオの進行上の矛盾も防げるしプレイヤーの「思い込み」による「ハマリ」は減ってくるだろう。

要は開発者はプレイヤーに対してゲーム本編には関係のないことで悩ませてはいけないということだ。もっとも，こういったことは開発者以外の第三者のテストプレイで感知しうる問題なのだが……。

ストーリーはADVGの顔

ADVGで一番重要なのはシナリオだ。話がつまらなければADVGなんて出来の悪い紙芝居よりたちが悪い。その昔X１用に「クリスタルプリズン」(BOTHTEC)というのが出ていたが，これは絵が決してほめられたものではなかった。一応「瞬間表示」(懐かしいネこの言葉）なのだがグラフィックツールのラインとペイントで描いたばっかりのような，原色メロメロの当時としても少し見劣りするものではあった。が，話，ストーリーが素晴らしかったのだ。

主人公は記憶喪失の状態で目を覚ます。ベッドのまわりから家族といい張る見覚えのない顔が自分を覗き込んでいる。自分は強盗に殴られ記憶を失ったと教えられるが……。こんな感じで物語が始まり，実は自分はとんでもない陰謀にはめられていると段々と気づきはじめ徐々に失われた記憶も蘇ってくる。そして最後には驚くべきどんでん返しが。今でも十分通用する家族サスペンスものですから，BOTHTECさんグラフィックとシステムを強化してX68000へ移植してみてはどうかな。

終わりに……

まあ，話も絵もよいに越したことはないのだが，これだけハード・ソフト技術が進歩した今でもこの両方のバランスが取れた作品は数少ない。そろそろ話のネタも切れてきたのでお勧めの最近作を挙げて終わりにしよう。

リバーヒルソフトの「J.B.ハロルドシリーズ」。第１弾の「殺人倶楽部」は話の展開と結末が少々アガサ・クリスティじみているが全体としては面白い。また，X68000はグラフィックが写真からの取り込み画像に独特の修整を施したものを使用しておりこれだけでも必見である。また，このあとの「マンハッタンレクイエム」「キス・オブ・マーダー」も水準以上の出来でお勧めだ。

同じくリバーヒルの「藤堂龍之介シリーズ」もお勧めだ。第１弾の「琥珀色の遺言」は洋館のなかで起こる連続殺人事件の解決に挑む探偵の活躍の物語だ。ソフトの値段は9,800円とちょっと高めだが値段相応の内容であることは私が保証しよう。

第２弾の「黄金の羅針盤」はまだXシリーズには移植されていないが，先日PC-9801で解き終わった感想をここで述べさせてもらえるなら「いいーっ！　いいーっ！うきっきー」のひと言（ひと言とはいわないかもしれない)。

こちらはサンフランシスコから横浜へ向かう豪華客船が舞台。ここである晩殺人事件が起こる。ここは海の上，船の上，犯人は同じ船の上にいる！　たまたま同じ船に乗り合わせた藤堂龍之介が捜査に乗り出すというお決まりパターンで物語が始まる。このゲームで私が心を奪われたのは，なんといってもグラフィック。アニメ風とは違った実写に近いタッチで描かれた登場人物はとてもゲームの雰囲気にあっていてお見事。ゲーム本編のほうは少しダミーメニューが多い感もあるが前作に劣らぬ完成度。

ADVGを語るうえでシンキングラビットを忘れたら犬神様の祟りがあるってなもの。ここはポリシーを持って今でも「コマンド入力」方式の作品を送り出す数少ないソフトハウスだ。代表作には「カサブランカに愛を」「ザ・マシ・アイラブ」がある。「コマンド入力」方式ではあっても各状況に要求される単語は難しくないので「言葉捜し」の心配は無用だ。しかし，ゲーム中「ハマリ」が随所に存在し，そのことはプレイヤーには知らされないので，そのへんは少しナニである。

そういえば最近はエルフソフトやフェアリーテールの出すいわゆる「Hソフト」のたぐいもレベルの高いものが数多く出てきている。「リップスティックADV」や「DE・JA」なんかは後半のムリヤリエッチシーンはともかく，立派な探偵もの，冒険ものとして結構楽しめる。

また，ダミーメニューに対する返事も遊び心溢れるメッセージが多く結構楽しい。「話す」，「机」と選ぶと普通なら「それはできません」と返ってきそうだが「リップスティックADV」では「机と話す…机と話す…俺は一体何を考えているんだ」と思わずニヤリとしてしまう台詞が返ってくる。また，電信柱や壁の貼り紙を「調べる」と乱数で主語，動詞を選んで文章を構成する「作文ゲーム」的なものが返ってくるなど，結構工夫がなされている。まあ，たまにはこういったジャンルのADVGを楽しむのもいいかもしれない。

最後までおつきあい有難う。今年はADVGが豊作であることを願いつつ……。

脳を刺激する隠れた要素

ゲームをつくる「質感」とは

Urakawa Hiroyuki 浦川 博之

ゲームをプレイするなかで，知らずしらずのうちに作られていくイメージ。絵の動きや音の作り方でプレーヤーのゲームに対する感じ方はずいぶんと変わってくるものです。そういう，感じ方がどこからくるものか考えてみてはいかがでしょう。

ほっほほほほ！　いきなり高飛車な笑いをとばしてしまう私だ。なぜって，ゲームに関する新しい見方に気づいてしまったからである。人の知らないことを知っているというのは気持ちがいい。世界征服を思いついた悪の帝王のような気分だ。ここに気をつけてゲームを見れば，なぜズームがあんなにウケるのか，なぜ気持ちいいARPGといまいちなARPGがあるのか，なぜビデオゲームの移植が「どこか違う」ものになってしまうのかが解明できてしまう。まさに1991年の激動のゲームシーンを語るのに欠かせないアイテムといえよう。ふっふっふ。

と，ひとりで悦に入ってるだけじゃ単なるやな奴だな。そのゲームに対する新しい見方というのは，

「ゲームの質感」

である。

何？　さっぱりわからない？　抽象的すぎるって？　よろしい。じっくり説明してあげましょう。

サークとラグーンと「質感とは何か」

たとえば，アクティブロールプレイだ。ここにX68000用のサークとラグーンがあったとする。両方とも上から見た構図のゲームで，戦闘は剣を持ってヤアと体当たりするという仕組みだな。

この2つ，同じスタイルではあるけれど，どこからどう見たって別のゲームだよね。グラフィックもサウンドも操作法もゲームシステムもみんな違う。

だけど，違うのはそれだけじゃない。言葉にするとかえってわかりにくいけれど，プレイしているときに感じるキャラクタの重さとか，キャラクタの材質なんかの違いがある。これが僕の考える「ゲームの質感」なのだ。

なんだ，そんなことかと思う人も多いと思うけど，実は結構重要な問題じゃなかろうか。敵にぶつかってスカッと相手をやっつけたと感じられるかどうかは，ゲームをやってるときの気持ちよさに直結しているから。

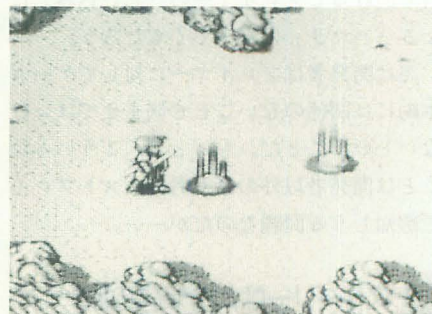
自動車の批評記事なんかを読んでいると，スタイリングや性能，居住性と並んで，必ずエンジンの吹け上がりがどうだとか，曲がり方が自然だとか不自然だとかいう話が出てくる。ゲームでの質感の高さというのは要するにこのエンジンのフィーリング云々に相当する部分なんだな。

そういう観点でラグーンとサークを比較してみよう。ラグーンのほうはスプライトを使ったスムーズな動きが特徴だ。戦闘の際にはダメージを喰らったほうが跳ね飛ばされるんだけど，そのときまですすーっと動いてしまう。サークだとチップ構成（ある大きさの部品で画面を作る方法）になっていて，ダメージを与えた瞬間，敵は2チップ向こうに一瞬にして移動している。

で，どっちのほうが剣で「突いた」感じがするかというと，これはサークのほうだ。こういうときはプレイヤーの目が追いきれない動きのほうが当たりの激しさが実感できるんだよね。ラグーンはなんだか寒天みたいなツルツルしたものがススッと滑っていくような感じがする。それに敵が死ぬときは跳ね飛ばされないでその場で消滅パターンが描かれるんだけど，これも非現実的な感触だ。狙ってこういう効果を出しているんなら脱帽ものだけど，プレイしている限りはそんな感じはしない。

一方のサーク。VRシステムというのを採用して画面の中になるべくリアリティを持たせるのがサークのコンセプトだけど，戦闘の感覚までリアルかというとそうでもない。というのは，スライムもスケルトンも同じ距離しか飛んでくれないからなんだな。サークではキャラクタごとに大きさがかなり違うので，同じように飛ぶと重さまでいっしょに感じられて奇妙なのだ。それにイースなんかだとぶつかった瞬間キャラクタ同士が重なるんだけど，サークはそうしていないので，お互いが固い材質でできているという感じがすごくする。

まあ，プレイしたことのない人にはわかりにくい説明だけど，ゲームによってキャラクタの重さや材質が違うというあたりは誰でも感じたことがあるんじゃないだろうか。

スプライトらしい動きのラグーンだが，戦闘の手応えが従来のARPGとだいぶ違う

PC-8801/9801などの機種で確立されたオーソドックスなグラフィック表示のサーク

ジェノサイドと「質感のしくみ」

で，このゲームの質感という考え方をもってすれば，ズームの人気を解明することができるのではないかと私は大胆にも考えた。僕は新作ソフト紹介の担当もしているので実際にいろんなソフトを試しているけど，ズームのゲームというのは質感においてはずいぶんほかと違うような気がする。特にそれを感じるのがジェノサイドだ。

ジェノサイドの質感といえば，なんといってもくすんだ色調で描いた背景とキャラクタ。キャラクタ自身の描き方なんかも結構独特なものがあって，いままでの色鮮やかなアニメ調とは違った質感が出てるよね。

だけど，一番独特なのはトレーサーが相手を斬りつける動きだと思う。ジェノサイドを買うような人にはバリバリのアクションゲーマーが多いと思うけど，あれだけ多くのゲーマーがジェノサイドをそろって支持しているのは，実はこの斬る動きの中に出てくる質感に原因があるんじゃないかと思うんだな。

ジェノサイドの面白いところは，アニメーションパターンの作り方にある。何もないところで剣を振り回すとやたら素早いんだよね。脳ミソは「うおっ，こいつは切れそうだ」という情報を受け取るわけだ。

これがいざ敵に出会った場合，剣が当たった瞬間にトレーサー（主人公ね）が一瞬弾かれるような動きをするようにプログラミングされている。しかもジャンプ斬りをしてたりすると，なかなか剣が下へ降りていかない。これがプレイしている人間には「剣は刺さったけど，いま自分はものすごく固いものを斬っているぞ」という情報になってフィードバックされてくる。ジェノサイドのキャラクタっていうのはどれも固いんだけど，単に何回も斬らなきゃいけないだけじゃなくて，感覚的にもすごく固いものを斬ってるぞという気がする。

ここがゲーマーにとって，実は非常に大きな魅力になっているんじゃないだろうか。敵の固さに感覚的な裏づけがあるからこそ，「うおお，ぶっこわれろぉ」と叫びながらジョイスティックのボタンを思いっきり叩けるし，相手が爆発したときの爽快感も味わえると思うんだよね。

プリンス・オブ・ペルシャと「質感の要素」

このゲームは質感がいいとか，このゲームはダメとかいってるだけでは単なる結果論だ。次に質感を高めるためにはどうしたらいいか考えてみよう。

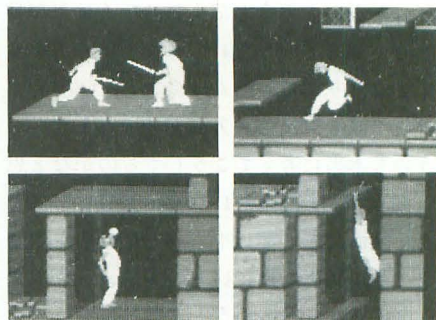
解答の一部は，これまでの中にもう出てきている。アニメーションパターンや動きのプログラムなんかがそれだ。

だけど，本当はもっと細かな部分が複雑にからみあってできている。だからいままであんまり明確に指摘されなかったんじゃないかな。たとえば，同じアニメパターンを採用しても，どのアニメーションを何秒表示するかによっても動きの感じは大きく変わるんだよね。

そのいい例がAMIGA版とPC-9801版のプリンス・オブ・ペルシャだ。この2つ，確かに同じゲームではあるんだけど，やってみるとぜんぜん違う。グラフィックが大幅に変わっていてその影響もあるんだろうけど，フィーリングの点でも同じ人間を操っているとは思えないくらいだ。

実は日本版のアニメーションは見よう見まねで移植したものらしい。その結果，なんか軟体動物っぽくなっちゃって，剣を振るときもシャープさがいまいち足りない。もちろん日本版だけをやっていればそんなことは感じないんだけど，AMIGA版をやってしまうと「何か違うぞー」という感じがする。

AMIGA版のほうは，さすがモデルさんが動くビデオ映像に合わせて作ったというだけあって人間のシャープな動きがよく出ているし，お互い剣を繰り出すと本当に丁々発止と渡り合っているように見える。外野の勝手な意見かもしれないけど，日本版もプログラムの移植が無理だったんなら同じビデオから動きを起こしてほしかったな。

音も質感の重要な要素だ。AMIGA版のプリンス・オブ・ペルシャの床の材質は何だと思うだろうか？　どうとでも受け取れるデザインだが，プレイした人にとっては，これは間違いなく「金属」なのだ。床が落ちるときにPCMで「ばちゃーん」という鉄板の落ちる音がするようになってるんだな。音が質感に与える影響はけっこう大きい。鉄板や石盤だと思うからこそ，頭にそれが

AMIGA版プリンス・オブ・ペルシャ。静止画で見るとラフな感じだが，ひとたび動くと実にいきいきとした本物らしい動きをする。見ていて不思議だ

落ちてきたときの痛さが実感できるし，プリンスがトゲにささるときに「ブスッ」という音がするから，見る人をあんなに痛がらせることができるのだ。

あともうひとつ，ゲームの質感を構成する要素として加えたいのが画面モード。ファンタジーゾーンは異なる画面モードで遊べるようになっていたけれど，15kHzモードのオパオパは31kHzよりもアラが目立つ分，1ドット1ドットがゴリゴリしたものに見える。ビデオゲームの移植はこのモードで動くものがほとんどだけど，やっていてビデオゲームとどこか違う感じがしちゃうのは，たぶんこの画面モードのせいなんだろうな。256×256ドットという独特のドット配分もキャラクタのデザインや動きに影響を与えているし，こりゃビデオゲームの移植を質感まで完璧にしようと思ったら，同じ解像度でやらないといけないということになっちゃう。質感ってやっかいな代物だな。

ゲームプログラムの質感

この質感というのは拡大解釈すれば，ゲームのプログラム全体にまで適用することができる。画面がパッと消えるのと，フェードアウトしていくのと，上から下へ消していくのとでは，違った感じをプレイヤーに伝える。ここがいかにもプログラムの都合っぽいと，せっかくのめりこんでいるのに途端に興ざめしてしまう。TVの映画を見てる最中に「次回予告」なんてテロップが入るようなもんだな。

なんか強引な論理だなという気も我ながらするけれど，こういう風にいわないと作り手も受け手もこういったことの大切さに

気づかないと思うんだよね。いまはただ目新しいからというだけで画面の消し方を変えてみたりするゲームは多いし，X68000ではいつでもフェードアウトしてしまうというのも本当に考えてこうしてるのかなと思っちゃう。ま，確かにフェードアウトはどんな画面にでも使いやすいんだけど，TVなんかで場面転換にいちいちフェードアウトを使ってるかというとそうじゃない。

あと画面が消えた瞬間に「白いカーソル」が点滅してしまうゲームもある。あれは質感以前にゲームの画面そのものが「作りもの」という感じがしちゃうからやめてほしい。劇の大道具を裏から見せるようなもんだもんね。

結論

この問題の難しいところは，どんなに質感にクセのあるゲームであっても，やりこんでいくうちに慣れてしまうというところだ。だけど，いつまでも不自然なゲーム感覚を押しつけてくる状態というのははっきりいって不健全だと思う。

ウィンドウシステムなんかもそうだけど，実際には存在しないけど頭の中では納得のいく動きってあるよね。現実にあるものの動作感覚を使って自然に操作することができる。ゲームもこれからはそこにこだわらなきゃいけないと思う。

特にドライブゲームのように何かをシミュレートしたものは，どれだけ感覚的に自然なものにするかということがゲームの中の大きな部分を占めていると思うんだよね。そういう意味では作り手もプログラムの都合をいつまでも優先させてはいられないはずなんだがなぁ。

パソコンゲームの世界ではずっと「ゲームのスピード」と「動きのスムーズさ」をどうクリアするかという問題がゲームの上にのしかかっていたけれど，ハードの性能やソフト技術の進歩でいまではほぼ解決している。それなのにゲームをやるほうの目はそのころからあんまり進んでないような気がするんだな。

単にスプライトを使って動きをスムースにすればいいというもんでもないはずで，いかにプレイヤーに自然な材質感を与えるかというのがこれからの注目点になっていいはずなのだ。

こんなことはいままではあんまりいわれなかったことだけど，これからみんなもっとこだわっていいんじゃないだろうか。シューティングブームとやらで似たようなスタイルのゲームが増えているけど，それらのゲームの人気を分けるのは，ラスタースクロールとかパワーアップなんかの目立つ技術じゃなくて，質感の完成度かもしれない。'90年代は質感を出したものの勝ち，なのだ。

ごめんなさいのコーナー

3月号「SXLIFE PartⅢ　ライフゲームで姓名判断？」で，リスト8がリスト7と同じものとなっていました。申し訳ありません。正しいリスト8を掲載します。

```
  1:         .even
  2: doMenu5:
  3:         bsr     _TmGetStr
  4:         tst.w   d0
  5:         beq     dom5Skip
  6:         lea     _len(pc),a2
  7:         move.l  d0,(a2)
  8:         subq.l  #1,d0
  9:         cmp.w   #16,d0
 10:         bcs     dom5Skp2
 11:         move.w  #15,d0
 12: dom5Skp2:
 13:         lea     _name(pc),a2
 14: dom5Loop:
 15:         move.b  (a0)+,(a2)+
 16:         dbra    d0,dom5Loop
 17:         move.b  #0,(a2)
 18: dom5Skip:
 19:         bsr     initMap
 20:         bsr     setName
 21:         bra     noAction
 22:
 23:         .even
 24: setName:
 25:         movem.l d1-d7/a1-a6,-(sp)
 26: *
 27:         move.l  wPointer(a5),-(sp)
 28:         .dc.w   __GMSetGraph
 29:         addq.l  #4,sp
 30: *
 31:         moveq.l #0,d3               ;; X
 32:         moveq.l #0,d4               ;; Y
 33:         moveq.l #0,d6
 34: snLoop:
 35:         lea     _name(pc),a0
 36:         move.b  (a0,d6.l),d0
 37:         beq     snExit
 38:         cmpi.b  #$20,d0
 39:         bcs     snConti
 40:         cmpi.b  #$80,d0
 41:         bcc     snL1
 42:         sub.b   #$20,d0
 43:         ext.w   d0
 44:         add.w   d0,d0
 45:         lea     __sj1(pc),a0
 46:         move.w  (a0,d0.w),d0
 47:         bra     snCmn
 48: snL1:
 49:         cmpi.b  #$a0,d0
 50:         bcs     snL2
 51:         cmpi.b  #$e0,d0
 52:         bcc     snL2
 53:         sub.w   #$a0,d0
 54:         ext.w   d0
 55:         add.w   d0,d0
 56:         lea     __sj2(pc),a0
 57:         move.w  (a0,d0.w),d0
 58:         bra     snCmn
 59: snL2:
 60:         addq.l  #1,d6
 61:         lsl.l   #8,d0
 62:         move.b  (a0,d6.l),d0
 63: snCmn:
 64:         move.w  d0,d1
 65:         moveq.l #8,d2
 66:         moveq.l #$16,d0
 67:         trap    #15
 68:         move.l  d0,a4
 69: *
 70:         moveq.l #0,d5
 71: sn2Loop:
 72:         movea.l a4,a1
 73:         moveq.l #$83,d0
 74:         trap    #15
 75:         movea.l a1,a4
 76: *
 77:         move.w  d0,d1
 78:         move.w  #15,d2
 79: sn3Loop:
 80:         btst    d2,d1
 81:         beq     sn3Skp
 82:         move.w  #15,d0
 83:         sub.w   d2,d0
 84:         add.w   d4,d0               ;; Y+15-K
 85:         move.w  d3,d7
 86:         add.w   d5,d7               ;; X+J
 87:         lsl.l   #6,d7
 88:         add.w   d0,d7
 89:         lea     _field(a5),a0
 90:         move.b  #1,(a0,d7.w)
 91:         move.w  d3,d7
 92:         add.w   d5,d7               ;; X+J
 93:         move.w  d0,-(sp)
 94:         move.w  d7,-(sp)
 95:         bsr     pset
 96:         addq.l  #4,sp
 97: sn3Skp:
 98:         dbra    d2,sn3Loop
 99:         addq.l  #1,d5
100:         cmpi.l  #16,d5
101:         bcs     sn2Loop
102:         add.w   #16,d4
103:         cmpi.b  #64,d4
104:         bcs     snL3
105:         add.w   #20,d3
106:         moveq.l #0,d4
107: snConti:
108: snL3:
109:         addq.l  #1,d6
110:         cmp.l   _len(pc),d6
111:         bcs     snLoop
112: snExit:
113:         bsr     drawGrowBox
114: *
115:         movem.l (sp)+,d1-d7/a1-a6
116:         rts
117:
118:         .even
119: __sj1:
120:         .dc.b ' !"#$%&'()*+,-./0123456789:;<=
>?@ABCDE'
121:         .dc.b 'FGHIJKLMNOPQRSTUVWXYZ[¥]^_'abc
defghijk'
122:         .dc.b 'lmnopqrstuvwxyz{|}~ ',$00
123:         .even
124: __sj2:'
125:         .dc.b ' 。「」、・ヲァィゥェォャュョッーアイウエオカキクケコサシス
セソタチツテトナ'
126:         .dc.b 'ニヌネノハヒフヘホマミムメモヤユヨラリルレロワン゛゜',$00
127:         .even
128: _name:
129:         .dc.b   '盛者必滅会者常離',0
130:         .even
131: _len:
132:         .dc.l   16
```

SOFTWARE information

スコルピウス

アーケードゲーマーなら一度はお世話になっている月刊ゲーメストの編集部有志が作ったというバリバリシューティングゲームです。横スクロール6ステージ，縦スクロール1ステージの構成。やられても復活できますが，難易度は低くありません。さすがにスーパーゲーマー集団が作ったゲームのことはあります。

サウンドはあの源平討魔伝やベラボーマンなどを手がけた中潟憲雄氏。これは期待大ですね。 (S.K.)

今月は軒並み発売の遅れるソフトが多くって困っちゃった。ホントならみんなの知ってるゲームの話がいっぱいできるはずだったんだけど。ごめんねぇ。来月はきっと紹介する，したい，できるといいなあ……。

話題のソフトウェア

まず発売中はスタークラフトの**クォータースタッフ**，M.N.MのMagical Shot，**アルガーナ**，ウルフの**哭きの竜**などなど。

開発中の目玉商品は，新声社の**スコルピウス**。これはあの月刊ゲーメストのお兄さまたちが作っているシューティングゲーム。かなり期待してよさそう。ほかは，日本ソフテックのeXOn。ライフゲーム風の，進化ものです。詳しくは次のページで。

開発が遅れながらも順調（なんじゃそりゃ）なのはブロダーバンドの**プリンス・オブ・ペルシャ**，T&Eの**遙かなるオーガスタ**，タケルの**ノスタルジア**。来月を待て。

ズームの**ファランクス**，工画堂の**サブナック**，ホームデータの**マーブルマッドネス**，NCSの**シグナトリー**もしゃかりき開発中。

さあ，来月こそはあったかくなることだし，どこかへ列車で行こう，と思いつつなんたら島，を夢見てみたいもんだ（意味不明？）。というわけで，またね～。

パロディウスだ! が登場前1位!

		(前回順位)
1	パロディウスだ！	2↑
2	エメラルドドラゴン	6↑
3	カオスの逆襲	－初
4	ソルフィース	1↓
5	シムシティー	5
6	イメージファイト	10↑
7	キャンペーン版大戦略	－初
8	ファランクス	－初
9	メルヘンメイズ	－初
10	ラプラスの魔	－初
	A列車で行こうIII	－初

たしかにGAME OF THE YEARのあとは得票がドドッと変わるのですが，先月5つも入れ替わっておいてまた新登場6つというのはすごい。では，今月もまた初登場作品の声を紹介しましょう。

カオスの逆襲：X68000を買ったら一番にやりたいソフト。一層奥が深くなっている。マウスをおシャカにするだけの価値はあると思う。

キャンペーン版大戦略：IIIがやりたいけどとりあえず発売になるキャンペーン版を推薦します。スピードと操作性のUPを買って。メガドライブの大戦略の上はいってほしい。

ファランクス：また楽しめそう。ズームだから。ズームだから。ズームだから。ズームだから。……みんな，ほかの理由も書いてくれよお。

メルヘンメイズ：かわいいから。でもムズそう。かなり前から移植してほしいと思っていた。ゲーセンの隠れた名作だと思う。

ラプラスの魔：とにかくハミングバードはいい。操作性・スピードがすこぶるよい。シナリオもよいけど，ムズい。ホラーは嫌いだが心霊ものは好きだ。

A列車で行こうIII：シムシティーより完成度が高いと思う。PC-9801版を見ていると最高に楽しそう。X68000でも遊びたいから。

まだ発売になってないうちからがんがんランクインさせちゃうなんて，みんな鼻息が荒いなあ。パロディウスだ！　なんて，私もまだソフトを見てないぞ。でもまあ，こうやってソストハウスにプレッシャーをかけるとさらにクオリティアップするかもしんないから，みんながんがんハガキ書いてね！　それじゃ，来月までさようならあ。 (浦)

＊画面は開発中のものです

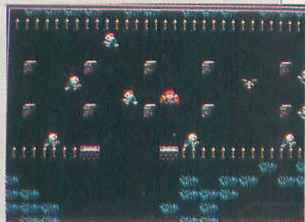
サブナック

アクションとパズルのよさを融合させた新しいタイプのゲームです。RPG風の画面の中で，呪いによって石化された妖精がどこかに転がっています。プレイヤーは石化を解く赤いマントを使って妖精を目覚めさせ，もともといたはずの台座に連れていくと面クリアです。

途中にはあなたの行く手を阻もうとする敵キャラが出現します。あなたは攻撃する武器を持っていないので，道に転がっている（？）戦士や魔法使いを赤いマントで目覚めさせ，彼等に戦ってもらうのです。ここいらへんがアクションパズルといわれるゆえんでしょう。

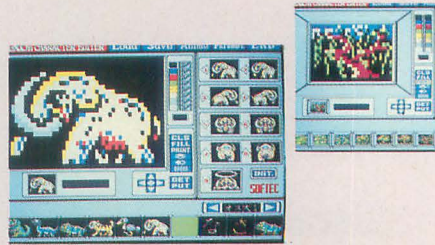
赤いマントの効果はプレイヤーのすぐ周りしかありませんので，障害物に引っ掛かって立ち止まった戦士や魔法使いは，その場で石になってしまいます。自キャラだけではなく，味方キャラすべてに気をつけていないといけないというのは，ボコスカウォーズを思い出してしまいそうですね。

プレイヤーの侵入を拒もうとする難解な迷路は全部で40面。地上から地下迷宮まで多彩な広がりを見せています。　(S.K.)

X68000用 5″2HD版2枚組　予価7,800円(税別)
工画堂スタジオ　☎03(3353)7724

eXOn（エグゾン）

新タイプのライフゲームです。プレイヤーは創造主となってコンピュータより早く，より繁栄した生態系を作りあげることが目的です。eXOnではマップは6時代あり，それぞれの時代上でゲームは進行していきます。

まずは1番下の時代で植物が数本と動物が数匹ずつ存在しています。そこから植物も動物も繁殖し，捕食しあい，進化・絶滅を繰り返しながら時代を渡り歩き，定められた進化を競争していきます。この間コンピュータが操る生物との戦いもあるようです。プレイヤーは自分の生物たちが全滅しないように気候の変化や繁殖期の設定などを変化させていきます。

ダーウィンもびっくりのキャラクターエディタがあります。ゲーム中に登場する植物6種，動物12種，特別キャラ6種をすべてエディットできます。進化の過程を自分で作れるわけです。さらにマップエディタで温度や湿度，登場キャラなどを各時代ごとに設定を変えることもできます。　(S.K.)

X68000用　5″2HD版　価格未定
日本ソフテック　☎0425(82)1502

シムシティー テレインエディター

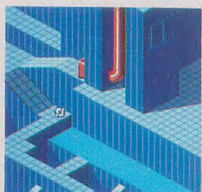
取り出したりますこのツール，テレインエディターという名のとおり，かのシムシティー専用のマップ作成のエディタでありまする。

最近シムシティー中毒患者から「もっとやりやすいマップを」とのお嘆きが聞かれますが，それは無理難題というもの。人によってやりやすさは千差万別，皆様お望みのマップを1つひとつ作っていたのではイマジニアが破産していまいます。

そこでこのテレインエディター。これを使用していただけば，お好みのマップがすぐさま作成できるわけでございます。また，一から作るのが面倒とおっしゃる貴兄のためにも，既存のマップをいじくりまわせる機能もついております。海を陸地に，街にスエズ運河を，なんてことも思いのままでございます。

とにかく，このツールひとむつで伸び悩んでいた街も復興，新しく作成したマップで心機一転，不満解消，家庭円満てなもんでございます。さあ，買った，買ったぁい！　(香)

X68000用　5″2HD版　4,800円(税別)
イマジニア　☎03(3343)8911

マーブルマッドネス

最新のサンプル版が届きましたので，先月に引き続きお届けしましょう。はやる気持ちをおさえて軽く説明してみましょう。ボールを転がしてゴールまで運んでいくという単純明快なゲームです。途中にプレイヤーを阻む仕掛けや邪魔者がいたるところに配されています。掃除機やスライムは序の口，降りづらい坂道やジャンプしなければ通れない道など，プレイヤー泣かせは無数に存在します。ボールでやるフィールドアスレチックといった具合です。きっとあなたもマウスやトラックボールを極限までいじめることになるでしょう。

実際に遊んでみた感じでは納得のいく仕上がりになっています。グラフィックの出来はAMIGA版を凌駕するものになっているし，SEもなかなか，肝心の動きもよくできています。

マウスでやるのは邪道なのかもしれませんが，トラックボールにしろ愛用の小道具は手入れが肝心です。　(S.K.)

X68000用　5″2HD版　価格未定
ホームデータ　☎078(261)2790

ノスタルジア

インテリジェンスアクションゲームと銘打ったゲームが登場します。サンプル版が届いたのでさっそくお知らせしましょう。

基本的に会話形式による1対1でのやりとりがあり，そこで得られる情報によってストーリーが進行していくようです。強気でいったり，弱気でいったりと，13種類の感情があって，その中からひとつを選択することによってセリフが展開していきます。そこで，こちらの感情によって相手の反応も変わってくるというものです。相手の性格を1人ひとり理解してないと聞き出せる情報も聞き出せなくなってしまうかもしれないのがミソでしょう。会話の部分だけはその昔にあったメイドゥムのような感じのシステムです。さらに相手の微妙な感情やニュアンスの変化を音楽が教えてくれます。

グラフィックはセピア調で統一されていて，なかなかきれいでしょう。

制作しているタケルはあのソフトベンダー武尊とは違いますので問い合わせの際は間違えないようにしてください。　(S.K.)

X68000用　5″2HD版　11,800円（税別）
タケル　☎03(3839)1013

Magical Shot

4年ほど前にビリヤードっつーもんが大流行りしてからというもの，コンピュータゲームにもビリヤードゲームが続々と登場。でも，どれもゲームにすぎなかったんだよね。理由はひとつ，視点の問題。どれも真上，真横からしか見られなかった。その点，このゲームは3D仕様。思う存分好きな角度に台を動かして遊べるのです。ビリヤードってのは，手玉をどの位置からどう打とうかって考えるのが楽しいんだもんね。

開発がちょっと遅れてたけど，いよいよ発売。このゲームで腕を鍛えていざ，ビリヤード場へ！　……行かないって。（香）

X68000用　5″2HD版　7,800円（税別）
M.N.M Software　☎0423(60)3084

クォータースタッフ

このゲーム，一応RPGというものの，よりAVGに近い。まあ，テーブルトークRPGはAVGにも近いものなので，そちらの方向に行くにしたがって，境目がなくなるのは当然か。ゲームブック感覚というのがいちばん妥当かと思われる。

雰囲気づくりに重点を置き，メッセージや部屋の配置など随所に気配りがうかがえる。それゆえ話はわりとありきたりのファンタジーだが，奥深さが感じられるものに仕上がっている。

その他の特徴：マウス操作（ショートカット可），プロテクトなし，ハード＆RAMディスクへのインストール可。親切なヒントあり。（R.A.）

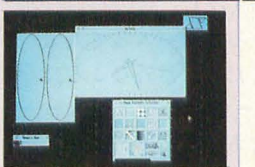
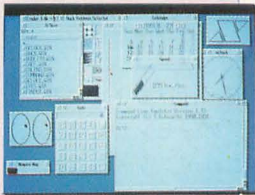
X68000用　5″2HD版2枚組　9,800円（税別）
スタークラフト　☎03(3988)2988

KO-WINDOW

あのCGAシステムのDōGAからオリジナルのウィンドウシステムが発売されます。すでに本誌でも何度か紹介されていますね。X68000のSUPER以外に付属のビジュアルシェルとは違って，SX-WINDOWばりのウィンドウシステムです。ビジュアルシェルとの最大の違いは，ウィンドウ上でファイルを実行できることでしょう。もちろん，マルチタスク処理になっています。サンプル版ではあの超有名な目玉や時計，電卓などは当然のごとく入っています。さらにはコマンドラインエミュレータというものがありました。ウィンドウ上で動くコマンドシェルのようなものですが，コマンドシェルに慣れた人には便利でしょう。ほかにもパブリスというテト○○に似ているゲームやネコなどもあります。

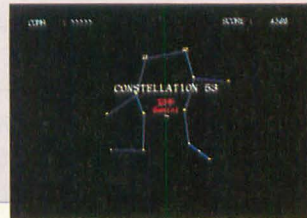
C言語で開発できる環境などを考えると，ウィンドウ上のアプリケーションを開発してみたい人など，かなりのパワーユーザー向けともいえそうです。メモリ1Mバイトでも動作するのは見逃せないポイントです。

なお，このシステムは現在ネットで流れているものと同じものです。（S.K.）

X68000用　5″2HD版　1,000円（税込）
ブラザー工業（TAKERU）　☎052(824)2493

COMET

このゲームはマウスで彗星を操作して星を囲んでいくゲームです。ステージは全天の88星座。美しい星座の世界が広がります。ゲーム内容が単純明快なだけに，独特の雰囲気の奥の深いゲームになっているようです。はじめのうちは止まっていた星たちが面を進めるごとに動きだしたり，時間がかかりすぎると邪魔者があらわれたりします。なかには暗すぎて見えない星があったりと，まるでプラネタリウムのソフトのようなディテールの細かさです。そうそう，いくつかの星をまとめて取ると高得点になります。

もともとは，かの祝一平氏の満開製作所に投稿されたゲームですので，電脳倶楽部にも掲載されていました。そちらのほうで手に入れた人も多いことでしょう。電脳倶楽部を買い損ねた人でも，おなじみのソフトの販売機にいけば手に入るわけです。（S.K.）

X68000用　5″2HD版　2,000円（税込）
ブラザー工業（TAKERU）　☎052(824)2493

（善）のゲームミュージックでバビンチョ

まずは，ごめんなさい。先月はスペースの都合で当初予定していた製品全部を紹介することができませんでした。というわけで，今月紹介するのはすべて発売中のものです。ご了承を。

●サウンドワールド・オブ・ポピュラス
ポニーキャニオン

あのポピュラスがCDに！　というと「どーせ，オーケストラかなんかに置き換えただけだろ？」とかいわれそうだが，今回のこのCDはちと違う。あの荘厳なメインテーマをアレンジしてボーカルを入れてしまったもの（！）を始め，プロミストランドをモチーフにした曲もあり，全体としてスマートな仕上がりになっている。また「大江戸編」では和楽器による変則ロックなどが入っている。しかし，三味線や琴の音に混じって，クイーカ（「できるかな」のゴンタ君の声のような音を出す楽器）やサンバホイッスルが鳴っていたけど，ありゃ和楽器じゃないぞ。
・勉強や読書のBGMにいいかもね。
お勧め度　7

●アクトレイザー/古代祐三
アルファレコード

まったく，この人のやることには毎回驚かされる。今回はスーパーファミコン用ソフトのBGMということで，同機が誇る音源，8チャンネルPCM音源を駆使した荘厳でシンフォニックな曲が全18曲収録されている。曲風は全曲管弦楽のノリで，従来のゲームミュージックに飽きていた人にはもってこいだ。それと，やはり全チャンネルサンプリングというのはいい，音が厚い。彼の持つ音楽的才能と高品質の音源が見事に融合した今回のアルバムは，日本のゲーム音楽史上屈指の出来栄えといっていいだろう。
・値段が2,000円というのは安い。
お勧め度　9

今月はこれまで

「アクトレイザー」のエンディングテーマは，映画通なら笑えます。「20世紀フォックス」のテーマにそっくりなのです。確か「フェリオス」のエンディングテーマも，これに似ていましたね。単に思いつきやすいフレーズを「20世紀フォックス」が使っていただけという見方もありますが……。それではまた来月。

再会はボクのX68000で……

Fukuhara Tohru
福原 徹

「不思議の国のアリス」を題材にしたアクションシューティングゲーム。3年ほど前にアーケードで登場，根強い「アリス」ファンの支持を得て，いよいよX68000に！　初心者でも楽しめる作品だ。

あたしが放浪の電脳絵師：福原徹(職業/古柴香織17歳）である。いやぁ，ひさびさにオモテ原稿が書けて嬉しいなぁ（うるうる)。長年，Oh!X編集部にはお世話になってるけど，いまだにゲームのレビューって書いたことないあたし……。いやね，文才がないから使ってもらえないってのもあるんだけど(苦笑)，あたしゃ基本的にゲーマーじゃないんだよね。どーもゲームやるのって面倒臭くてね。最近はめったにジョイスティックを握らない。

アクションゲームっていうと，シューティングドンパチドンパチなのが多いじゃない？　あれってみ〜んな同じようなシチュエーションに見えるんだよね。新鮮さが感じられない。日本漫画のルーツが手塚治虫だったように，アクションゲームのルーツってゼビウスだったような気がする。いまだにみんなその影響引きずってる。あたしゃとっくにゼビウスで果てた。

これってきっと「愛」なのね◆◆◆◆◆◆

さてさて，今回SPSより発売されたメルヘンメイズは，1988年の夏ごろ登場したナムコのアーケードゲームからの移植だ。当時，某パソコン誌のアーケードゲームコーナーにメルヘンメイズの紹介記事が載ったときは，運命の糸を感じちゃったね。なんてったって「詩とメルヘン」愛読者だからね僕は(うそ)。こういうゲームに注目しないわけがない。決して新しいコンセプトを持ったゲームではないんだけど，グラフィックのセンスのよさと，登場キャラクターのかわゆさにひかれて，気がつけばゲーセンに足を運び，普段ほとんど座ったことのない筐体の前へ……。

いや〜，アリスちゃん，

「めんこい！」

しかし！　やっと4面へ行けるようになったかと思ったらすぐ，ゲーセンから姿を消していたんだよね，これが。うぅ，なんていけずぅな店（涙ナミダ)。

やっぱこーいうナンパなゲームって人気ないのかなぁと思ってたら，意外なことにX68000に移植されるってニュースが入ってきた。嬉しかったなぁ。待っててよかった。ゲーセンから消えてしまったあのときは，そりゃ絶望のどん底で，いっそ自分で移植でもしてしまおうかとさえ考えましたよ，真面目に(実力がともなえばネ)。こんなマイナーなゲーム，まっさか移植してくれるメーカーさんがあるとは思ってなかったからね。そんなわけでさっそく編集さんに「紹介記事書かせてください！　ワクワク」攻撃に出たわけです。

そうしてみると，案外このゲームのシンパって多かったみたいですね。移植が実現したのもファンの要望が多かったからでしょうし，事実，この記事を書く権利をめぐっては，僕と山田氏の間で熾烈な争奪戦が行われたのでした(笑)。むふふ，根回しの成果だよ，純ちゃん。

2年半振りの再会だね◆◆◆◆◆◆◆◆

ゲームの内容は，メルヘンチックなモトスってな感じでしょうか。ようするに敵キャラと主人公アリスが，空中のクォータービュー・ステージで落としっこをするわけだ（そう考えてみるとアリスちゃんっては力士みたいだな)。ただし，自分から体当たりしても，ちっちゃいアリスは敵にはね飛ばされるだけ。彼女の唯一の武器は「シャボン玉」(うぅ，なんて健気なんでしょ)。これを相手に吹きつけて落とす，もしくは破壊する。

このシャボン玉には，パワーが数段階あって，小刻みにBボタンを押せば連射ができるが，それは小さくて破壊力は弱い。Bボタンを押し続けることによってシャボン玉は大きく膨らみ，その破壊力が強まる。ただしその場合もちろん連射はできない。また押し続ければ無限に大きくなるというわけではなく，一定の大きさをこえると壊れて消えてしまう。敵の強さと自分の状況を的確に判断してタイミングよくシャボン玉攻撃すべし！

なお，転がってくるボールや小さい敵はAボタンでジャンプして避けられる。制限時間内でボスを倒せばその面はクリアだ。

X68000用　5″2HD版2枚組　8,800円(税別)
SPS　☎0245(45)5777

アイテム紹介

ステージ上のいたるところに「？」の書かれた箱がある。これを壊すとさまざまなアイテムが出現する。これを使えばアリスがパワーアップするぞ。

★赤玉　ボーナスポイント	★箱　アイテムそのほか（？）が入っている
★青玉　一定時間だけ最大パワーのシャボン玉を発射できる	★時計　制限時間を延長してくれる
★黄玉　敵の動きを一定時間止める	★風船　落下してもその面に限って一度だけ救ってくれるぞ
★緑玉　3方向へシャボン玉を発射できる	★くつ　ジャンプ力がアップして、遠いところまで飛べるようになる
★紫玉　アリスの足が速くなる	★うさぎ　「うさぎさんバリア」となって一定時間アリスの周りをまわって守ってくれる

うさぎが誘うポエムな世界 ◆◆◆◆◆◆

真夜中にふと目覚めてしまったアリス。なにやら話声が聞こえてくる……。

「だれ？」

「たすけてください！」

鏡の向こう側……うさぎだ！

「私たちの国がなにものかにしんりゃくされてしまいました」

「なぜ私が？」

「私たちと話ができる人しか，この世界に入れないのです。さ，早く！」

「!?」

こうして，なかば連れ去られるようにして，アリスは鏡の向こうの国を救うために旅立ったのでした。

ステージは全部で９つの国。では，各面を簡単に説明していこう。

★１面：おかしの国

最初の面だけあって，それなりに簡単。敵キャラはバキュラのように回ってくる鏡や，オパオパをさらに丸っこくしたようなロボット。煙草をくわえた生意気なキノコなどなど。近接信管つきのドクロ玉にも注意しよう。近づくと破裂する。この面のボスは「魔法使いのおばあさん」だ。ちいさな子分を放出してアリスちゃんをいぢめてくれるぞ。

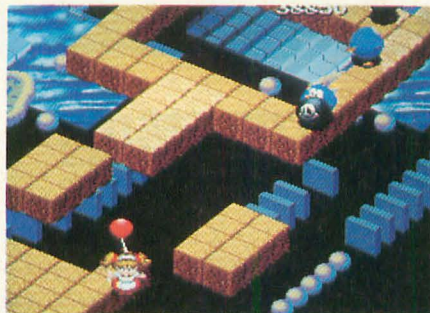
★２面：おもちゃの国

おもちゃのペンギンさんや玉乗りをするデブったおじさんが登場。ボスはゼンマイ仕掛けの「鳥のおもちゃ」だ。ボスの吐き出すボールの数にはムラがあるので，薄いところからねらって攻撃しよう。

敵を蹴散らす！　うさぎさんバリア

１面のボス，魔法使いだ

風船を持っていてよかったネ！

４面の主戦力，ロボットだ

★３面：つみきの国

ここはジャンプのタイミングが勝負を左右する。失敗すると上下運動する壁にはねとばされて，まっさかさま。個人的にはいちばん苦手な面。ボスは「ダルマおとし」。下の段からひとつずつ地道に破壊していこう。

★４面：きかいの国

ドクロが出てきたりして，見掛けはオドロオドロしくて難しそうだが，案外と簡単。雰囲気は男の子向きだね。ボスは古典的な「ロボット」。

★５面：そらの国

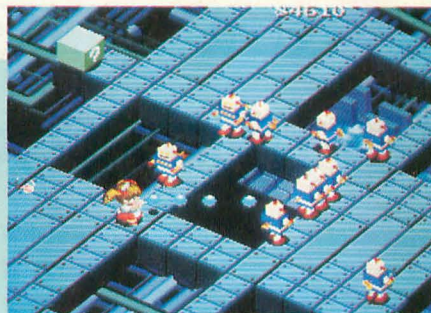
トンボやヒラメの舞い踊りが見られるぞ。足元のブロックの動きがフェイントっぽいので注意。ここは「甲殻ザメ(?)」みたいなのがボスだ！

★６面：びっくりの国

この面はやたらと敵が多くて大変。中華風所ジョージみたいなのと，牛の顔した敵がいるぞ。こいつらの脚が速くててこずらされる。ボスは「かめ風船」。こいつは着地しているときにしか攻撃が効かない。

＊

以下７面，８面……と続くのだけど，ここから先はお楽しみということで，ヒ・ミ・ツ！　自分でプレイしてちょうだい。

総じてみると，ボスは比較的弱い。一般的なシューティングゲームのようにボスで進行を阻まれるということがあまりないように感じられる。落ち着いて攻撃すればそれほど怖いことはない。とにかくゲームクリアには根性が必要。あせらずにじっくり敵と戦い進め！　健闘を祈る！

出来はほぼ完璧,うん満足! ◆◆◆◆◆◆

先月号の紹介記事でも書かれているが，画面上のキャラクターが増えると，若干処理が重くなるようである。だが，ゲームをやっているときには僕はあまり気にならなかった。そのほか，移植の出来はほぼ完璧といっていいと思う。デモからオープニングに至るまで，アーケード版のクオリティを保った完成度。さすがはX68000，さすがはSPSである。原作ファンはひと安心だね。

それから，残念ながらこのゲームも要2Mバイトだ。いい飽きた感もあるが，1Mのシステムの人はこの際頑張って増設してしまおう。その代わり，オンメモリだから一度ロードがすんでしまえばディスクは抜いても大丈夫。

メカメカやエログロがゲーム業界に横行するなかで，あえてポップ&コミカルなデザインでユーザーを攻めた，心をなごませてくれる数少ないゲームである。女性ウケもかなりよいようだ。ゲームマーケットの未開の市場，「女性」を開拓するためにも，デザイナーの方々にはもっともっとセンスを磨いてもらいたい。

いやぁ，それにしてもアリスちゃんの

「あん！」

がひたすらカワイイ！

「かわいいカルチャー」とゲームの融合

登場時にはゲーセンで閑古鳥を鳴かせていた（少なくとも僕の周りではそうだった）のに，移植の要望では妙に盛り上がってしまったという不思議なゲームである。なんだか，その昔バンダイから発売されたミンキーモモのフィギュアの騒ぎを思い起こさせる（強引かな）。アリスフリークは熱狂的な人間が多かったということか？

それにしてもSPSさんは，いつもいつも僕の趣味に合ったソフトを開発してくれて，ホント感謝してます(へこへこ)。これからも期待してますよ。

当然ですが僕は買います！

総合評価	0　　5　　10
ゲーム性	★★★★★★
移植完成度	★★★★★★★★★
グラフィック	★★★★★★★★★
ミュージック	★★★★★★★
メルヘン度	★★★★★★★★
「あん！」	★★★★★★★★★★

(善)の中華大仙料理講座

CHUKA TAISEN
中華大仙
TAITO

「中華大仙」というゲーム知ってますか？　知ってますよね。誰にでもとっつきやすいというのが特長というシューティングです。見かけは「西遊記」そのものという感じもしますが，あまり関係ないようです。はい。

Nishikawa Zenji
西川 善司

ある大阪人の会話，

A：おい，このせんべい食ってみい。

B：ガリッ。

　なんちゅうかたいせんべいや！

と，いうわけであのタイトーの「中華大仙」がX68000で蘇った。「中華大仙」といえば1988年夏にアーケードに登場して以来，MSX2なんかにも移植されたりして，よく名前を耳にはした。でも，本物をやったことがあるかと問われると，結構みんなダンマリだ。私もゲームセンターで見かけた記憶はあるが，当時はパスしてしまった。しかし，アーケードでヒットしなくても家庭用ゲーム機などに移植されてから人気の出たゲームだってある。たとえば，「サンダーブレード」や「ニュージーランドストーリー」「熱血高校ドッジボール部」などがそうだ。だから，ゲームを評価する際に，アーケードでの評判を参考にするのはいいが「うのみ」にしては危険かもしれない。

SPSの歩み◆◆◆◆◆◆◆◆◆◆◆◆◆

話が「中華大仙」から10kmくらい離れてしまったので60cmほど戻すとしよう。

皆さんご存じのとおり，SPSの移植技術は目覚ましい進歩を遂げてきた。いまだからいえることだが，初期には失敗作と思われるものもあった。そして，全国のX68000ファンに数mlの冷や汗と涙を流させたのであった。「『グラディウス』のSPSはいまどこに」とまでいわれたものだった。しかし，その後の「スーパーハングオン」や「サンダーブレード」ではX68000のハードの限界を超えたとも思える完成度で，一気に汚名を返上し「技術のSPS」の名を不動にしたのであった。最近発売された「サイバリオン」，自社発売のナムコ「ワールド」シリーズなどはもう「お見事」のひと言につきる。とりあえずこれから先は，SPSが移植するアーケードゲームに失敗作はありえないといえるだろう。

今回発売された「中華大仙」もオリジナルそっくりの出来栄えとなっているのでゲームセンターで熱中したファンは安心して購入するといい。もっとも，この「中華大仙」，中古基板のほうも値段が安くなっていて1万円前後で買えるので，その筋の方はそっちもどうぞ。

「中華大仙」は単純シューティング◆◆◆

プレイヤー扮するは，見習い仙人マイケル・チェン（私は孫悟空かと思っていたのだがそうではないらしい）。マイケルってことはもしかしたら中国系アメリカ人2世ってことかな。どうでもいいかそんなことは。チェンは仙人の最高位である「中華大仙」の称号を得るために師匠より与えられた5つの試練に挑んだ……というのが大まかなストーリー。

それでゲーム内容だが，いたって単純。別に土地を造成して人を繁栄させなくたっていいし，監督になって甲子園を目指す必要もない，まして全国統一をする必要なんかまったくない。「中華大仙」は襲いくる敵キャラを撃ち落とすだけでいいのだ。そして，面の最後に控える本ボスを倒せば1面クリア。全5面だから，要するに，この動作を5回繰り返せば全面クリアということだ。難易度はそれほど高くないし，完全なパターンゲームなので練習さえつめば誰でも1周できるだろう。そう，「中華大仙」は，まさにプレイヤーを選ばないシューティングゲームなのだ。まあジョイスティックくらいはないと，ちときついかもしれないが……。また，連射の苦手な人には「連射機能つきジョイスティック」の購入をお勧めする。

ゲームシステム紹介◆◆◆◆◆◆◆◆◆◆

さて，撃つだけとはいっても独自のパワーアップシステムもあるので紹介しよう。

特定の敵を倒すと漢字1文字のアイテムが出現，これを取ることによって自機のショットや移動速度を上げることができる。

もうひとつ，自機をパワーアップする方法がある。面の途中に登場してくる中ボスを倒すと，イベントワールドと呼ばれる扉が出現する。これはなんと「ファンタジーゾーン」のショップのようなもの。ここに入ると数種類ある「法術」のうちひとつがタダ，つまり「無料」でもらえちゃうのだ。この「法術」とは早い話「特殊弾」のことで，以後やられるまで常時発射可能となる。また，その「法術」を身につけた状態で（つまりミスしないで）次の「イベントワールド」へ入ることができれば，より強力な「法術」をもらうことができる。法術は全部で

X68000用　5"2HD版　7,900円(税別)
シャープ　☎03(3260)1161

人面鳥来たり！　"力"でパワーアップして応戦だ

不良大仏，万里の長城に現わる

13種類，どれを選ぶかは君の自由だ，「have one's choice」ってなもんだぜ，ベイビー。

さあ，自機をパワーアップして「中華大仙」を目指せ。

最終面攻略◆◆◆◆◆◆◆◆◆◆◆◆◆◆

「中華大仙」は操作も簡単だし，コンティニュー回数も∞なのである程度の面までは進めると思う。しかし，途中でミスをして装備をなくすと復活が少々きつい。おそらく復活のきついのは，3面の後半と最終面の後半。紙面の都合により3面のほうはパスしてここでは最終面の後半の復活パターンを「ゲーメスト」風に解説しよう。図を見てほしい。

まず，スタートしたら自機を画面中央に持っていき麻雀牌を迎え撃つ。先撃ちして必ず全滅させること。と，岩が降ってくるのでこれを避けつつ下へ移動。画面下でドラゴンを仕留めると「力」が出現。これを取ったと同時に画面右下部より餃子の編隊がやってくる。自機はすでに画面下にいるはずなのでこれをすくい上げるように撃ち倒す。全滅に成功すればここで「強」が出現（これを取った時点で青の弾が2連装になっているはず)。

ここで気づくと中ボスのライオンが浮いている。ギリギリいっぱいまで寄って連射。オニギリが後ろからやってくるので気をつけて。図の注釈のあたりで中ボスは1回目の攻撃を仕掛けてくるが，中ボスへの攻撃は続けつつ後ろに下がりながら避ける。このときのタイミングは微妙で，慣れないうちは図中注釈のラーメン部隊にやられるかも。絶えまない攻撃を中ボスに浴びせ続けることができたプレイヤーは注釈の時点までに中ボスを倒せるはず。

このときのイベントワールドでの私のお勧めは「火竜」。見た目はパッとしないが通常弾の数倍の威力が魅力。本ボスもこれならラクチン。

とはいうものの，先はまだ長い。イベントワールドのあとには猫の生首軍団のお出迎え（猫生首は図にあるとおり長期にわたって飛来するので要注意)。このあとに餃子部隊が来る。これを全滅させると例によって「力」が出るがすでに「青弾2連装」の人はこれは取らない。理由は次段階の弾は連射ができないから。

次に来るのは「パンダ要塞」。画面左端に来たときの攻撃には注意。餃子の編隊がこのあと2回来るが「力」は先ほどと同様の理由で取らない，「速」は取ること。

そうこうしているうちに2匹目の中ボスがやってくる。「火竜」をもっている人はものの数秒で倒せるはず。「火竜」を本ボスまで持っていきたい人は中ボス破壊後のイベントワールドに入る必要はない。

このあと，地面に大砲豚が出現しはじめるので「火竜」の人は貫通性を利用して突破せよ。人面鳥が顔を見せはじめたら本ボスはもうすぐだ。本ボスは画面左下を中心に避ければ楽勝。でも，途中で「速」を1個は取ってないときついかも。

ちなみに本ボスを倒したあとは簡単なエンディングメッセージ表示後1面に戻りますです……はい。

というわけで西川善司の「中華大仙」講座，終わり。

ちょいといわせてもらえば……

出来はいいと思います。でも，いわせてもらえるなら，せっかくシャープがバックについているならもうちょっとメジャーなものに挑戦してくれてもいいと思うんですけどね。

たとえば，カプコンのCPシステムシリーズ（1941，LOST WORLD，マジックソード，大魔界村etc)，ナムコのシステムⅠシリーズ(ベラボーマン，妖怪道中記，スプラッターハウス，ロンパーズetc）やセガのシステム16シリーズ（ダイナマイトダックス，ゴールデンアックスetc）など。アイレムのM72にもおいしそうなのがあるし（最後の忍道，ドラゴンブリード，Xマルチプライ)。そうだ，ちょいとよくばってセガのシステム18，「シャドーダンサー」なんて私のお気に入りなんですけどね，とかいってこりゃマイナーだったか，けへへっ。

総合評価	0　5　10
グラフィック	★★★★★★
音楽	★★★★★
移植完成度	★★★★★★★★★
熱中度	★★★★★

Oh! ジャパニーズ・パズルゲーム

Yamada Junji
山田 純二

遅ればせながら，スライスの登場です。基本的にはコラムスと同じくブロックを揃えて消すのですが，これはブロックを横に倒すこともできるのです。音楽は古代祐三氏を起用，M.N.M会心の一作です。

は～，やっと出た出たついに出た，ってなもんで，「スライス」の登場だ。最初に見たサンプル版からキャラクターも一新され，背景のグラフィックも美しく生まれ変わり，おまけに古代祐三氏のBGMまでついちゃって，ずいぶんと成長したものである。

どこがスライスなの

基本はコラムズと同じで，上から降ってくるブロックをうまく組み合わせて，縦横斜めに3個以上同じブロックを揃えて消すだけ。ブロックが最上段まで積み上がるとゲームオーバー。コラムズと同じく，連鎖反応によるスコアアップが非常にカイカン。「か～っ，やめられまへん」てなもんだ。

「じゃあ，どこらへんが“スライス”なの？」という素朴な疑問に対しては，「ブロックを横に倒してスライドさせることができるのが“スライス”なんだよ」と，答えておこう。あに？　わからんとな。

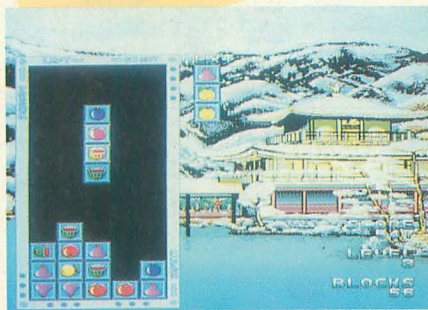
えっとだな，コラムズでは，落下中にボタンを押すことによってブロックの順番を入れ替え，組み合わせを作っていく。が，スライスでは，ブロックが落ちてくる間は左右にしか動かせない。落下中のブロックが別のブロックに対して接したときに，初めてプレイヤーの脳みそをゲシュタルト崩壊させるのだ。つまり，着地した瞬間に左右のキーを操作して，横にスライドさせたり転がして逆順にしたりするわけだ。

和風の背景にファンク，このミスマッチがいい

図1。この図は着地した瞬間に右のキー操作を行ったものだ。横に90度倒れているのがひと目でわかる。図2は，この状態から何も操作しなかった場合。今度は，180度反転してブロックが逆順になっている。これは左右に倒したとき，“落ちてきたブロックの数－1個”の空白があるとこうなる。ない場合には，それぞれのブロックが分離して下に落ちていくことになる。図3は横に倒した瞬間，左にキー操作した場合。見事にスライドしてるでしょ？　この状態からさらに左にスライドさせることも，元の状態に戻すこともできる。で，この状態でブロックが落ちると……。左右のブロックが分離して落ちていくようすが想像できた方，よくできました，それが正解です。

とまあ，結構操作が難しく，見かけにだまくらかされてコラムズと同じような感覚でプレイするとパニクッてしまう。ひょっとしたら，コラムズに慣れきったプレイヤーを陥れようとするデザイナーの悪意かもしれない。う～ん，ずっこいヤツ。

ファンキーじゃ

ちょっとだけBGMについていうと，ファンキーです！　古代祐三氏が担当していて聞いててちょっと楽しい。出だしの「ワンツーごしゅごしゅヘイ」が笑える。ほかはまあ，普通のBGMって感じかな。

ゲームの全体的なイメージとしては，ごくありきたりのパズルゲームといわざるをえない。アイデアもコラムズの亜流だしね。それでは，最後にいつものセリフで終わるとしましょう。

パズルゲームというのは一発アイデア勝負だから，もっともっと奇抜なアイデアが出てこないもんでしょうかねえ。

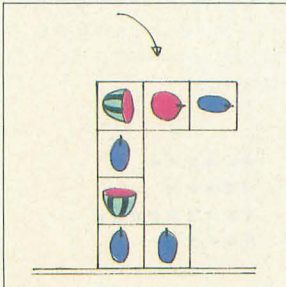
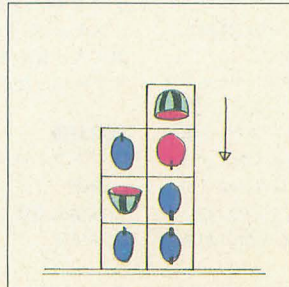
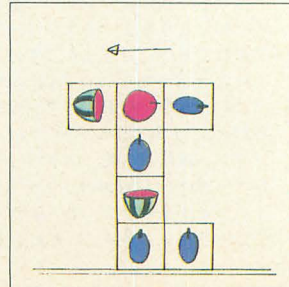
ランキング登録もあるでよ

さて，このゲームはM.N.M自社ブランドのパッケージで発売されている。ということで，値段のほうは決して“お手軽”という価格ではない(とはいえ，それほど高くはないが)。これを決めるのは，読者の皆さんにおまかせするとしよう。

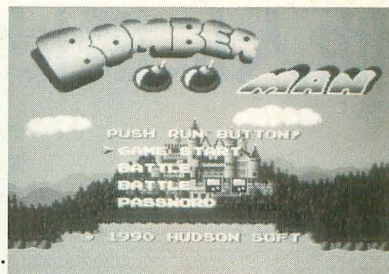
が，ゲーム自体はそれなりに楽しめる。スコアのランキングもディスクに保存されるので，友達なんかとワイワイ騒ぎながらハイスコア合戦するとさらに面白くなるであろう。

総合評価 (0—5—10)	
BGM	★★★★★★★★
グラフィック	★★★★★★
ランキング	★★★★★★★
お買い得度	★★★★★★★

X68000用　5″2HD版　7,200円(税別)
M.N.M Software　☎0423(60)3084

図1
図2
図3

爆弾男再び!?

数々のゲーム機に移植されている，古典的ゲームのリメイク版。爆弾を置いて道を開き，爆風に敵を巻き込んで全滅させる，というアクションゲーム。操作も簡単で，誰でも遊べるお手軽な作品です。

Yoshida Kenji
吉田 賢二

このゲームは，X1やMZ時代からある古典的なものなわけですが，当時は「ボンバーマン」ではなく「爆弾男」で売りに出されていました。「津村順天堂」が「ツムラ」になったのと同じく，最近は「漢字」の名前がカタカナ，アルファベットにリネームされることが多いようで……。

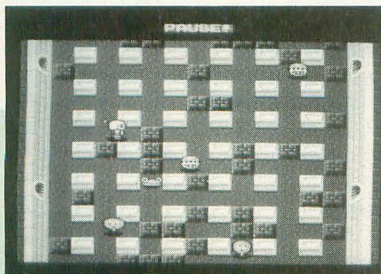
新「爆弾男」と旧「爆弾男」◆◆◆◆◆◆◆

さて，ディスクを起動すると美しいタイトルが滑らかなスクロールで登場。雰囲気はほとんどタイトーのキャラクターゲーム，これならゲーセンにあっても見劣りしないでしょう。つまりこれは最近流行の，歴史あるオールドゲームをメジャータッチでリメイクしたもの，なのです。では，ここでオールド版との相違点を挙げておきます。

・画面が格段に美しくなりグラフィックも可愛らしくなった
・ストーリーらしきものがつきオープニングやボスキャラがついた
・アイテムで自キャラのパワーアップが可能になった
・最大4人までの対戦モードがついた
・爆弾を半ずらしでおくことができない

つまり，

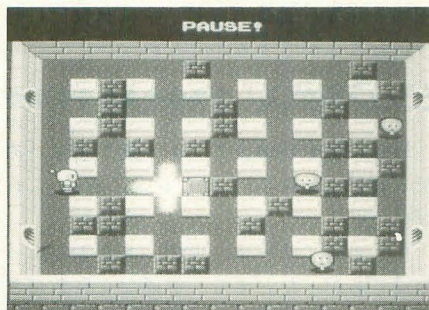
■■■←壁
爆弾→●
■■■

といった芸当が禁じられた。これによって戦略の立て方が前作とはかなり異なってくるのでチェック。

X68000用　5"2HD版　7,800円(税別)
システムソフト　☎092(752)5262

「爆弾男」を知ってるかい ◆◆◆◆◆◆◆

画面からも察しがつくとおり，主人公は4方向へのみ移動可能。トリガを押すと主人公の足元に爆弾が置かれます。これが数秒後に爆発すると，爆風が4方向へ走ります。この中へ敵を巻き込めば得点となり，敵を全滅させ脱出すれば1面クリアです。

新「爆弾男」では，爆風で吹き飛んだ壁の焼け跡から歩く速度が上がったり，爆弾の火力が上がったりといったパワーアップアイテムが出てくることがあります。これらは，各面1個ずつ隠されています。

面は6ワールドの8ステージといった構成で，ワールドが進むにつれて背景やモンスターも，刻々と変化していきます。また旧「爆弾男」は固定1画面ゲームでしたが，新「爆弾男」は縦横にスクロールする面もあったりとなかなか驚かせてくれます。そして各ワールドの最終ステージつまり8ステージ目にはボスが登場します。

ゴールを爆破すると敵が増えちゃう

ゲームオーバー。コンティニューコードが下に出る

さあ，あなたも私も爆弾男，笑う声まで同じねってな感じで（©サザエさん）最終面目指してガン，ガン，ガンバ！

ひー，ひー，いわせて♥ ◆◆◆◆◆◆◆

なんといっても楽しいのが対戦モード。キーボードでは2人まで，ジョイスティックをもう2本持っていればなんと同時に4人で競い合うことができるのです。対戦モードで倒すべき相手は対戦相手が操る「爆弾男」。また出現アイテムは「火力アップ」と「一度に置くことのできる爆弾の数アップ」（説明的でスミマセン）の2種類のみなのですが出現数のほうは本編よりもずっと多い。よって対戦の序盤は壮絶なアイテムの取り合いとなります。自分の出したアイテムを取られたり，自分の仕掛けた爆弾の爆風に巻き込まれ自滅してしまったり，もう大騒ぎの連続。何回戦もやっているとアイテムを通称で呼ぶようになってきまして，Oh!X編集部のマシンルームからは毎日のように「火ー火ーひーっ」「たまたまたまーっ」といった奇声が聞こえてきます。

さあ，予備校探しに忙しい浪人生のあなたも卒業できずに留年が決定した大学生のあなたも，今日はいい目を見ている友人を家に呼び込んで爆弾男大会でもどうです？

気付いた点を幾つか

画面が切り替わるときの待ち時間が長い。なんであんなに待たされるんでしょう。あと，対戦モードで3人以上でプレイした場合，初めに死んだプレイヤーがとても暇になってしまう。死んだらモンスターを操作できるとか，ゲームには直接関与できない幽霊キャラとなって他プレイヤーの撹乱ができたりするとよかったのに。なんにせよ，ひさびさの家族みんなで遊べるゲームですよ，これは。ひーひー。

総合評価	0　5　10
キャラクター	★★★★★★★★★
サウンド	★★★★★★
ゲーム性	★★★★★★★
熱中度	★★★★★★★★
お買い得度	★★★★★★★
対戦の盛り上がり	★★★★★★★★★★

※このレビューは2/24現在のサンプルでプレイしたものです。

THE SOFTOUCH

●Ryu[哭きの竜]より

他人に授ける運など持たぬ

ウルフ・チームから麻雀ソフトが登場。しかし，ただの麻雀ではない。「哭きの竜」という麻雀漫画をゲーム化したもので，他の麻雀ソフトにない斬新なアイデアが導入されている。君は「Ryu」になりきれるか？

Kageyama Hiroaki
影山 裕昭

たとえばパチンコ。プロになれば釘を見る目はかなりのものだ。それでも負けることがある。運も勝敗を左右するからだ。これが麻雀となると，パチンコ以上に運の占める割合が多くなってくる。素人の強運が玄人の実力に勝つことだってある。

“運も実力のうち”という言葉がある。実力と強運を持ち合せた雀士がいたら，それこそ最強の雀士に違いない。

ゲームデザイン◆◆◆◆◆◆◆◆◆◆◆◆◆

このゲームは近代麻雀という雑誌に連載されている『哭きの竜』が原作になっている。竜は渋い。普段は無口な男だが，たまに口を開けばクサイ台詞を吐く。「あんた，背中が煤けているぜ」なんてのは名言だ。かっこいい。

ウルフ・チームが麻雀という思考ゲームを作った。しかし，いままでの傾向からいって，ありきたりの麻雀ゲームを作るとは思えない。開始直後，いつもほどの派手さはないが，雰囲気たっぷりで，見ずにはいられないデモが流れる。渋い音楽をバックにして，まるで映画の予告を見ているみたいだな。このへんはウルフだな，やっぱり。

ゲームモードは3つ。お好みの面子で遊びたいなら，フリー雀荘モード。僕は麻雀の道に進んで飯を食おうというほどの者ではないので，コンピュータの思考力を見究めるのは難しいが，まあ普通に麻雀を楽しめるレベルに仕上がっているといえる。

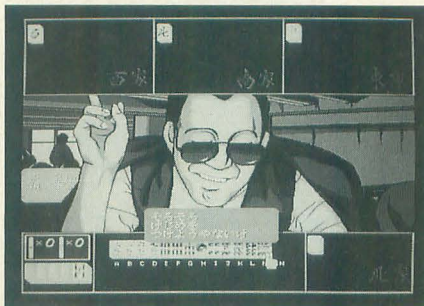
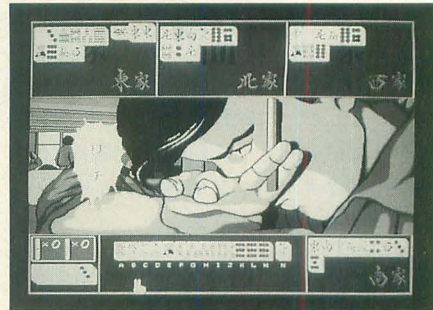
めずらしく思ったのは，他家の手牌がゲーム中はまったく表示されず，捨牌だけということ。他家の手牌が表示されていると相手がつもった牌を手に入れたかどうかがわかるのだが。考えすぎかもしれないが，竜が麻雀を打つスタイルをゲームに表現したら，こんな形になったのだろうか。

さて，このゲームのウリは2つのドラマモード。まず，自分が甲斐組組員「外田」となって竜を捜すというもの。麻雀で勝って竜の手掛かりを聞き出していく。もうひとつは記憶をなくした麻雀打ちとなって，自分が竜であることを証明していくもの。特に後者は自分のツキがよくなってくると，面白いようにほしい牌をつもるし，竜になりきって哭き麻雀を展開して大役をあがったりすると（「牌に命を刻んでいく……」とか呟いたりはしないが），気分は実にいいものだ。自分が竜を選択したときは思い切って哭き麻雀を展開すると，ツキを自分に呼び込むことができるかもしれない。

X68000用　5"2HD4枚組　11,600円(税別)
ウルフ・チーム　☎03(5273)4795

ビジュアルがいい◆◆◆◆◆◆◆◆◆◆◆

最高に従来の麻雀ゲームと違う点は「あがりビジュアル」と「役ビジュアル」。これがいいぞ！

たとえば，リーチ，平和，ドラ1といった役であがったとする。普通なら点数がさっと表示されて次の局に進むところだが，これは違う。画面中央にデカデカと「リーチ」と表示されると，リーチ棒のグラフィックが，ガラスの割れたような効果音とともに表示される。間髪入れずに平和と表示され，今度は太平洋の日の出を思わせる画面が表れる。最後に「ドラ」と表示，巨漢デブがドラを叩いている画面が表れる。

サングラスが怖いおにいさん

リーチをかける姿もキマっている

これがあっという間のことである。周りにいたスタッフも声を出して笑ってしまった。どうやら役ごとに，役名からイメージされるグラフィックを表示してくれるようだ。漫画の雰囲気がよく出ている。このグラフィックはON/OFFの切り替えができるが，全部の役を見るまでは表示ONにして遊んでみたいものだ。

怖いおにいさんがいっぱい

竜をとりまく人物像をゲームから感じとることは難しい。なによりも人間関係がまったくわからない。しかし，原作を読んでいると，哭きの竜の世界にすんなりと入ることができる。特にドラマモードでは原作を知るか知らないかで，熱中度がかなり違ってくるはずだ。実際，単行本を読んでから遊んでみると，数倍面白く遊べた。音楽の完成度も高く，ステレオにつないで聴くとさらに美しいサウンドが楽しめる。値段は少し高めだが，「哭きの竜」ファンなら買ってみて損のないソフトだといえる。

総合評価 (0〜10)	
グラフィック	★★★★★★★★★
効果音	★★★★★★★★
感情移入(予備知識有)	★★★★★★★★★★
感情移入(予備知識無)	★★★★★★★

リングの騎士よ，先は長い

Urakawa Hiroyuki
浦川 博之

ボードゲームの大手メーカー，ホビージャパンが「テーブルトークRPGの雰囲気を，完全にコンピュータ上に再現！」と銘打つ。「リングマスター」シリーズの第2弾の登場だ。さて，どこまで再現されているかな。

どんなゲームなわけ?

このゲーム，「テーブルトークの雰囲気を再現」というのがウリ。で，どのへんがテーブルトーク志向なのかというと……。

まず，前作同様の人工無能と呼ばれる会話ルーチン。入力した内容に対し，それらしく答えてくれるというものだ。そんなに頭はよくないけど，これのおかげで一応テーブルトークっぽい気分になれる。

次に世界設定。これはなかなか凝っている。この世界に関するガイドブックがついていて，プレイヤーの住む国フィリアス・ノギス，今回の舞台となるムルソン公国などの地理，歴史，宗教，政治，自然などがバッチリ載っているのだ。背景世界がきっちりしているので，村人との会話やゲーム中の記述に深みが出ていて，違う世界にトリップしてる感じが味わえる。

最後にパラメータや判定の計算式が明確であるということ。行動や戦闘の成功率などの計算方法を明記している。テーブルトーク出身だと，こういうところをしっかりしたくなるんだろう。パソコンゲームでは因果関係のわからない行動やパラメータが多いので，これはいいことだと思う。

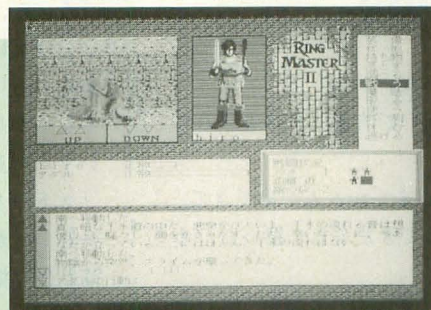
また，テーブルトークとは関係ないが，特筆すべき点としてプロテクトがかかっていない。バックアップを取ることもできるし，DOSの知識があればハードディスクへのインストールも可能だ。そのかわりマニュアルプロテクトというのがあって，細かい字でちまちまと書いてある数字を入れなければならない。ゴトランタ地方のクイズにでもしてくれればよかったのにな。

で，やってみると

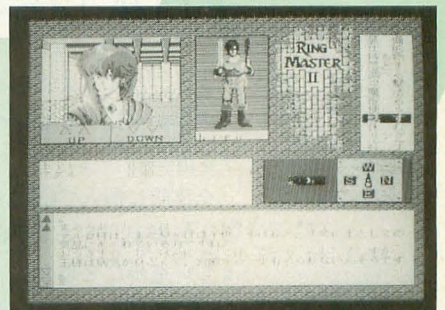
まず，プログラム的に遅い。ディスクアクセスに時間がかかるのはハードディスクなりなんなりに転送すればいいとしても，マップを描くのが遅いし，テキストも表示が遅いのでふにゃふにゃっと出てくる感じがする。反応にシャープさがほしい。

マップの表示にしても，ほとんどPC-9801版のデジタル8色で描いた画面のまま。ブルトン・レイではそれなりに情緒があったけど，下水道や洞窟のグラフィックは地形が把握しづらいのでちょっとね，という感じ。それに迷路にアクセントがないために地理が覚えづらい。画面の使い方もいまいち。装備が目に見えるのはいいけど，自分以外のメンバーは表示されない。昼と夜を示すゲージも小さくて見づらい。

だが，シナリオには合格点がつけられる。フィリアス・ノギスの王様に，ムルソン公国への援軍要請という使命を与えられるのだが，到着してみるとムルソン宮廷は王子が誘拐されたと大騒ぎの真っ只なか。援軍を出すどころではないと聞いて，プレイヤーも王子を探しに出ることになるというわけ。事件の全貌が明らかになるにつれて，さまざまな背景とかが見えてくる。ほかの国により道するとショートシナリオも用意されている。

シナリオがいいからこそ，しっかりしたシステムで遊びたかった。テーブルトークのノウハウは豊富だけど，コンピュータゲームとしてまとめるノウハウが足りなかったということかな。もっともっとコンピュータゲームを作っていけば，いずれ解決する問題ではあるんだけどね。

おなじみスライムが襲ってきた

会話を楽しんでみる

X68000用 5"2HD版3枚組 8,800円（税別）
ホビージャパン ☎03(3354)9341

パソコンとシステムのマッチングに課題

いや，惜しい。ゲームをしながらこの世界のガイドブックをぺらぺらめくってみると，話が見事に一致しているところには惹かれるんだけど，ともかくプログラムやシステムがついてきていない。

アクション型とならび，テーブルトーク志向というのもRPGの新たな活路のひとつになるものだと思うので，パソコンにうまく馴染ませることができれば，とても面白いゲームになると思うんだけど。この路線は本腰を入れて気長に熟成させるべきだと思います，はい。

総合評価 (0～10)	
操作性	★★★★
音楽	★★★★★
グラフィック	★★★★★
ゲームデザイン	★★★★★★★★
プログラム	★★★
熱中度	★★★★★★

CGAコンテスト座談会

プロジェクトチーム DōGA かまた ゆたか

先日行われた「CGAコンテスト入賞作品上映会」には，皆さん足を運ばれましたでしょうか。さて，今月はCGAコンテストについて審査員その他もろもろの方々にお話をうかがいました。できればビデオを見ながら読んでいただけると，いっそうよくわかると思います

CGAコンテスト特別企画第2弾ということで，審査員の方々に，お話をうかがってみました。また，ビデオ配布のお知らせもありますので，お見逃しなく！

はじめに

隔月連載だから，今月はお休みのはずですが，CGAコンテスト特別期間中（？）ということで，おじゃまさせていただきます。

以下は，審査員の方々を中心に，当チームのスタッフや，ただの通りすがりの方まで，入選作品をご覧になった方々の意見を，座談会ふうにまとめたものです。最初は実名で行おうとしたのですが，どの審査員がこの作品についてどう言ったというのは，あまりおおっぴらにするべきではないという見地と，カセットテープに録音したのをあとで聞いてみると，どの声がどの人かわからなくなってしまったという理由（実はこれだけだったりする）から，すべて匿名とさせていただきました。

CGAコンテスト座談会

「うん，みんなようやっている」

「去年から比べるとレベル高いですね。どれもきれいですよ」

「全体的に個人でCGAをここまで作る人が出てきて，ここまで表現できるようになってきたことに驚いた。めざましい進歩だと思う」

「自主制作っていうものが，最近8㎜フィルムとかがなくなってきたせいか，だんだん減ってきた。それが，こういうところで立ち上がりつつあるというのがとてもうれしい」

「コンピュータによって，個人では難しかったアニメーションというものが，結構簡単になってきた。コンピュータの技術的な進歩と，個人の新しいものを作りたいという欲求がうまくかみ合っていますね」

・お行儀のいい作品

「ただ，笑いが少なくなった，というか，全体的にまじめに作っていますね」

「Oh!Xの連載で，アニパロはやめようと強く言ったので，その影響かな？」

「でも“アジオージャ”（注：昨年度の入賞作品で，アニパロ色が強かった）みたいなのが，ひとつ2つ混ざると光るんですけどね（笑）」

「アニパロとか，自分の世界に遊んじゃっている作品ばかりだと，ちょっと辛いから，これでいいんじゃないですか」

「う〜ん，でも今回はみんな文学的というか，お行儀のいい作品ばかりという感じはする。もっと，コンテストあらしというか，はちゃめちゃなものもあっていいんじゃないかなぁ。そんな作品があれば，賞をあげる……というわけじゃないけど（笑）」

「ストーリー性なんかが重視されて，設定とか準備段階からしっかりやっているって感じだなぁ。まぁみんな作品制作に慣れてきたんでしょうね」

「でもそれが，アマチュアっぽい型破りというものを弱くしてる。自分で枠をはめている，プロの映像の真似をしているって感じ」
「漫画でも，映画でも，最初は人の真似から勉強するってことは，よくあるんじゃないですか。でも確かに，賞をとることを意識して作っているという感じはありますね」
「でも，来年はもっとはちゃめちゃなのも期待するっていうと，はちゃめちゃなのばかり集まってしまいそうで怖い（笑）」
「むちゃくちゃばからしいのがあってもいいと思いますよ」

・長いことは美徳ではない

「逆に，ストーリー性がもっと強くて，20分とか，30分とかそういうレベルで見られる作品が出てきてもいいんじゃないかな。今回，そんなレベルの作品が出てくる可能性というものを強く感じた」
「そうですね，レイトレースで作った，一瞬きれいなだけって作品なんて，もう耐えられない。ストーリー性があって面白ければ，多少ガタガタしていても長時間見られる」
「どうですかね。今回の作品を見て，強く思ったのですが，長いってことは美徳じゃないですよ」
「そうそう，今回の作品は長すぎる。2/3，いや半分にできる作品が目立つ」
「やっぱり作者自身は，あの長さでいいと思っていらっしゃるようですね」
「僕らがTVアニメを作る場合でも，自分が描いた原画なんかは，少しでも長く見せたいものですが，演出家にいわせると，もう少し見たいな，もう少し見せてくれたらな，というところで切っちゃうのがプロなんです」
「CGじゃなくても，自主制作の作品はそういう傾向がありますね。ここは苦労したんだから，できるだけ長く見せたいってのはわかるんですが，見る場合は，凄くマイナスですね」
「うん，いいたいことをまとめて短くしてほしい」
「でも，CGでよく見かけるんですけど，やたらに短いカットがパッパッと切り替わって見づらいってのはいけない」
「いや，それは話が違う。単にカットを短く，作品を短くしろっていうんじゃない。30秒以上の長さのカットがあっても，もちろんいい。ただ，今回の作品は，10秒にすべきカットを30秒にして，30秒にすべきカットを1分にしているということだ」
「その点，上位入賞の“SWORD”や“CLOCK”は，いい編集ですね」
「それにはちょっとワケがあります。まず“SWORD”ですが，もともとSRANIM，つまり秒10フレームでアニメーションの予定だったのが，最後のほうでHANIM，つま

審査員紹介

◆矢野　直明
「ASAHIパソコン」編集長。1966年朝日新聞社社会部入社，「朝日グラフ」の編集者を経て，1988年「ASAHIパソコン」創刊より現職を担当。

◆土田　米一
「ASCII」編集長。アスキー社にて，同誌の副編集長を経て，1986年より現職を担当。昨年より「Super ASCII」編集長を兼任。

◆前田　徹
「Oh!X」編集長。ソフトバンク社にて，シャープ系パソコン専門誌「Oh!MZ」編集長に引き続き，現職を担当。

◆片桐　淳一
美術専門誌「イラストレーション」編集長。ビデオ雑誌，写真雑誌の編集者を経て，昨年より現職を担当。

◆井上　伸一郎
アニメ情報誌「NEW TYPE」編集長。1980年よりアニメ雑誌の編集に携わり，1985年より「NEW TYPE」の編集を担当。昨年より現職を担当。

◆寺島　令子
ファミリーほのぼの4コマ漫画家。1979年「少年マガジン スペシャル増刊」にてデビュー。現在「LOGIN」にて「墜落日誌」を連載中。
代表作：「くりこさんこんにちは」ほか。

◆加瀬　政広
アニメーター，作画監督。19歳のときよりアニメ関係の仕事に従事。代表作：テレビアニメ「ミスター味っ子」（キャラクターデザイン，総作画監督）。現在，オリジナルビデオ「スケ番刑事」を制作中。

◆塚田　哲也
CGデザイナー，CGイラストレーター。1987年CGプロダクションJCGLに入社，現在はフリーで活躍中。CMの制作や個展を開くなど精力的な活動をしている。
入賞歴：小学館　写楽　審査員特別賞ほか。

◆岩井　俊雄
映像作家。1985年ハイテクノロジーアート展ほかに入賞。映像をテーマにコンピュータやビデオを用いて，CM，プロモーションビデオの制作を行っている。現在はTV番組「アインシュタイン」のCGを担当。

◆古川　タク
アニメーション作家，イラストレーター。1970年代後半よりコンピュータを使ったアニメーションの制作を始める。代表作は「驚盤」。現在，岩井氏と共に「ASAHIパソコン」誌上にて「電脳絵師養成講座」を連載中。

◆鎌田　優
プロジェクトチーム DōGAチーフスタッフ。1986年，プロジェクトチーム DōGAを設立。CGA制作も行っており，1988年全日本ビデオコンテストほかに入賞。

CGAコンテスト事務局より

昨年ご好評いただきましたビデオ配布を，今年も行います。この連載の読者の方はぜひ申し込んでください。昨年は，60分ビデオで2,000円でしたが，今年は90分で2,000円のまま。できるだけ多くの方々にご覧いただけるように，一層努力させていただきました。内容的にも，絶対満足していただける自信があります。CGAファンのあなたには必須のビデオです！

名　　称：第3回　アマチュアCGAコンテスト　入選作品集
形　　態：VHSビデオ　90分
内　　容：入賞8作品，入選6作品，参考入選2作品（一部）
　　　　　そのほかCGA入門講座など
期　　間：1991年3月15日～5月31日（当日消印有効）
申し込み方法：郵便振替　口座番号　大阪3-109598
　　　　　加入者名　DōGA
配布価格：2,000円（送料込み：カンパ任意）
発　　送：1991年6月下旬（予定）
注　　意：・払い込み用紙には，住所，氏名，連絡先の電話番号を明記してください
　　　　　・通信欄には，ビデオ希望と明記してください
問い合わせ：〒553　大阪市東淀川区淡路5-17-2　102号
　　　　　プロジェクトチーム DōGA内　コンテスト事務局

り秒20フレームのアニメーションに変更されたとおっしゃってました。ですから，作品としては，本人が最初計画していたテンポより，早くなったんじゃないですか」
「つまり，作者がいいと思うテンポの倍ぐらいにすると，見る人にとっては適当ということになりますね」
「そこまでいい切る自信はないですけど（笑)。“CLOCK”の場合，監督と制作者がまったく異なって（先月号参照)，ある程度客観的な目で編集できたという点が大きいんじゃないですか」
「そういう制作方法は有効だと思いますよ。映像作品である以上，見る人の目を持たないと」

・映像作品として通用するか

「そうそう，CGっていうのは，なんていうのかな，環境ビデオに近くなっちゃいがちなんですよ。最初から最後まで，目を離さずに見るという感じより，その雰囲気を楽しむって作品が多い」
「少なくとも映像である以上，目を離さずに見られる，というより，目を離させない演出が必要だと思う。えっ！と驚くようなシーンとか」
「そういう意味で，グランプリの“SWORD”はりっぱな映像作品として通用する」
「ついに，パソコンだからとか，CGだからという言い訳なしに，作品だけを見せて通用するものが出てきたということですね」
「うん，“SWORD”だけでなく，“CLOCK”だっていい線いってるし，“ORIGIN”や“帽子屋さん”なんかも，NHKなんかでそのまま上映しても恥ずかしくないと思いますよ」
「いままでは，パソコンで絵が動かせるだけで面白がっていた。やっとその領域から出たって感じ」
「そうはいっても，映像作品として一流だと胸を張れるようなレベルでは全然ない。プロの映像作家の作品の中には，たった15秒でも人を感動させるものがある。ああいったものと比べると，差は大きい」
「そりゃ当然でしょう。でも，プロの中でも超一流の作品といきなり比較するのはちょっと酷だと思いますよ」
「まだ生まれたばかりのメディアなのだから。実際にいま取り組んでいる絶対数，層の厚さの問題もある。だから，ここ数年の成長性や将来性に注目したい」
「各自がもっと映像の勉強をしていく必要があると思います。いろんな意味で，表現力がまだ足りないように思う」
「注意しないといけないのは，見る側の人にとってはその作品だけが評価の対象だということ。つまり，パソコンで作ったんだとか，自作のプログラムで作ったんだといった背景まで考えない。いきなり，TVのCMや，劇場の映画と比較する」
「どうかな？　このコンテストは，映像のコンテストとは違うわけだし，CGAっていうのは，映像だけでなく技術の分野でもあるわけだし。その作品の背景というものも評価に入れていいはず」
「このコンテストの開催趣旨を考えると，どうやって作

柚姫の明るい悩み相談室

ちょうどこの号が出るのは，ちまたではホワイトデーが終わった頃かな。1カ月前のあの戦場のような2月14日に比べれば，騒いでいるのはごくわずかという感じ。姫も，お返しって特にもらったことないなぁ。

姫の小学生だった頃の話なんですが，お返しにも，その中身によっていろいろと意味があるそうです。まず，自分も相手の子に好意があるとき，これはホワイトチョコかマシュマロを返す。次に友達だというとき，これはクッキー，そしてどうしても好きになれない場合は，塩せんべい（！）なんだって。これって，大阪だけかなぁ（実は姫の住んでいるとこだけだったりして……)。

ま，とりあえずお便りいってみましょうか。

＊

Q：CGAシステムの配布についてですが，もう手に入らないのでしょうか。また，展示会や，コンテスト会場などでのシステム配布はできないでしょうか。
姫：今はもう基本的には配布していないみたいです。
かまた：いや～，していないわけじゃないんだけど，やっぱり，今のバージョンというのは，根性がないととても使いこなせない。あんなバージョンを誰かれかまわず配布していると，CGAってやっぱり難しいんだという誤解を広めるだけじゃないかっていう気がして，ちょっと控えてるんだ。そういうわけで，根性はあるから，どうしてもCGAしてみたいという人には，まだ配布してもいいですよ。
姫：Ver.3が出たら広く配布するんですか？
かまた：Ver.3の話はやめてくれ～。とりあえず，手直ししたバージョンを，年内に配布しようと思ってるけど，マニュアル作り直すだけでも大仕事だからね。ちなみに，某L○G○N誌では，タケルで入手できると書いてあったそうですが，たんなる間違いです。
姫：ということで，どうしてもいますぐ欲しい人は，申し込んでみてください。あとはもう，あなたの祈り次第。心を込めて，大阪の方角に向かって1日五回，五体倒地して祈ってください。そのとき重要なのは，あなたの願いのひたむきさ，そしてなにより笑いを取ろうという気持ちです。オチが取れるかどうか，ここが運命の分かれ道です。
Q：つい最近になって，CGAシステムを知ったのですが，もう配布はされていないのでしょうか。もし配布されていないのなら，有償でも結構ですから，譲っていただけないでしょうか。
姫：あの～，なんか誤解していませんか？　このCGAシステムは，もちろん営利目的じゃないんですけど，無償で配っているわけじゃありません。マニュアルの印刷代，発送料などの実費とカンパで，一応4,000円以上いただいています。わかっていただけました？
Q：アニメーションを作ってみたら，動きが少しおかしかったんです。こんなとき，RENDでワイヤフレームだけでカットを作るようなモードがついていたら，動きが短時間でチェックできると思うんですが。
姫：うん，まったくだ。確かに「できた～！」と思ってみてみたら，かっこよく飛ぶはずの飛行機がなんと前転していたときのショックは……。だけど，人に頼ってばかりいてはいけません。自分で作りましょう。そう，あなたが作るんです，がんばってね！
Q：FFE, ATRがうまく使えません。FFE, ATRの使用についてのコツを教えてください。やっぱり努力と根性でしょうか。
姫：ずばり，愛情でしょう。日々変わらぬ，愛情を注ぎ，昼夜を問わず傍らにいて話しかけてやってください。それでもなつかないような親不孝者の場合は，センジンの谷にでも突き落としてやってください。はい上がってきた子だけが，あなたのCGAシステムです。

＊

姫は，水曜日の授業を選択しておらず，お休みにしています。ある水曜日のこと，姫がいつものように，のんびりと休みを満喫していると，チャイムが鳴りました。玄関で待っていたのは，“エ○バの○人”というある宗教団体の人でした。低血圧で頭の回っていなかった姫は，聞かれるままに，毎週水曜日は休みであることを答えてしまいました。……その後，その人たちが毎週やってくるようになったのです。誰か，なんとかしてくれいっ！

ったかというのも評価の対象になるんじゃないかな。アマチュアCGAなんて始まったばかりで，その制作手法なんかも確立されたものじゃない」

「そうですね。ちゃんと技術賞なんかあるわけですから」

・次回のコンテスト

「それでは最後に，次回のコンテストに期待するものについて，お聞かせください」

「CGって，CMとか店頭で流れているだけっていう使い方をされているって気がするんですよ。素人の作品っていうのは，そういうのじゃなくて，いいたいことがいえる，やりたいことがやれるっていうとこがあると思うんです。ですから，一般的にあるやつと違うところを目指してほしい」

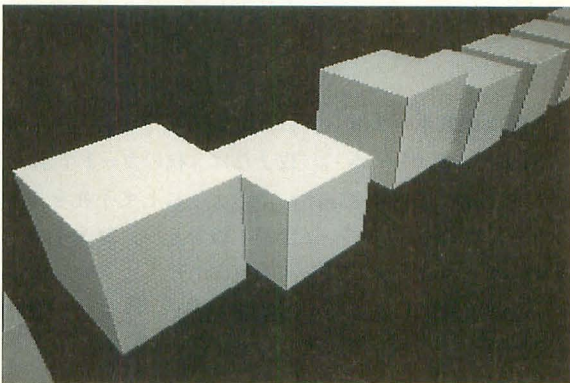
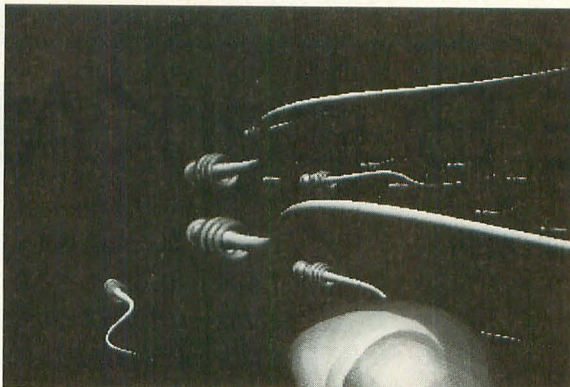
「そうですね，ここが見せたいというものを持ってほしい」

「どんなことに使えるのかっていうのを見つけていくことが大切でしょうね。そういう意味で今回CMや教育というジャンルを開拓していこうという作品があったのはよいと思う」

「ロボット物は，これからも一定量出てくると思うのだけど，動きのデザインというか，質量感というものを表現してほしい。モデリングだけではだれも驚かなくなってしまったから」

「ストーリー性なんかは改善されつつあると思いますが，いまひとつ，キャラクターが描けていないと思うのです。だから感情移入できない。登場人物，べつに人物でなくていいけど，キャラクター性にも注目したいな」

「全然別の話になりますが，今回CGAシステムばっかりで面白くなかったので，いろんなソフト，特に新しいアルゴリズムとかを使ったオリジナルのシステムなんかを期待します」

「なんか，ぜんぜんまとまってませんね(笑)。まあ，どのような作品でもよいから，どんどん応募してくださいということで，座談会を終了させていただきます」

おわりに

審査員の皆さん，お忙しいところ，ご協力くださいまして，まことにありがとうございました。次回もまた，よろしくお願いいたします。

さて，今回は，コンテストの準備が重なって，なかなか大変でした。ということで，来月こそはお休みにさせていただきます。次回は，のびのびになっていた，本格的モーションデザイン「戦えロボット君２」をやりたいと思います。かなり高度な話ですから，「戦えロボット君１」の前後の復習をしておいてくださいね。

ただの雑談 ［CGAコンテストでは，CGAシステムが有利か?］

今回のCGAコンテストで，ひとつ気になっていることがある。応募作品の８，９割がCGAシステムを使用しているということだ。そりゃ，CGAシステムがほかの（市販＆自作）システムより圧倒的に優れているのならわかるが，とてもそうとは思えない。もしかすると，当チームが主催しているコンテストだから，CGAシステムを使っている作品が有利だなんて思われているのかもしれない。はっきりいっておくが，そんなことは絶対にないぞ！

一般にコンテストなどにおいて，本当に“厳正なる審査”が行われているかといえば，かなり疑問だ。しかし，考えようによっては当たり前で，たとえ開催趣旨が“パソコンの文化の発展”と書いてあっても，実行事務局が，たとえば“パソコン販売促進部”であれば，そのメーカーの販売促進という目的を持った審査になるのは当然だろう。つまり，そのコンテストの目的が審査を“厳正”でなくしているのだ。

それじゃ，やはりCGAコンテストではCGAシステムが有利なんだと思うかもしれないが，それは違う。ご存じのように当チームの目的は，「手軽で，パーソナルな映像表現としてのCGAの普及」である。CGAシステムのPRとか販売促進などを目的にしていない。だいたい，自らCGAシステムの配布に制限をつけているのに，PRや販売促進して何のメリットがあるんだ？　CGAの普及という目的から考えると，いろんな表現力を持ったいろんなシステムがあったほうが好ましい。

もう一度はっきり書く。CGAコンテストで，CGAシステムが有利になるなんてことは，絶対にない。それでも，信じられないならもっと正直にいってやる。強いていうなら，CGAシステムでないほうが有利になる可能性がある。

なぜなら，使用したソフトが自作したものであれば，当然“技術賞”の対象となる。また，それが市販ソフトの場合でも，今回のようにCGAシステムによる作品ばかりのなかでは逆に目立つので，審査員に印象づけられる（かもしれない）。

ということで，次回のコンテストの応募に，CGAシステムでないからという理由でためらっている人を見たら，安心して応募するよう勧めてください。

X68000CARDDRV用カードゲーム

THE SuperKABU

Okubo Akihiro 大久保 明弘

今月はちょっと毛色を変えて，あのオイチョカブをカードゲームで再現しました。もちろんコンパイラにも対応しています。なお，このゲームを実行するには1991年1月号で発表されたCARD2.FNC（またはCARD.FNC）が必要です。

入力方法

日本伝統の由緒正しき賭博遊戯オイチョカブを基にしたCARDDRV用ゲーム“THE SuperKABU”です。元々は花札を使ったゲームですが，トランプのカードで代用することもできます。ちなみに「オイチョ」が8で「カブ」が9の意味で，集めた札の値が9に近づくようにするのがゲームの目的です。

CARDDRVを組み込み，CARD2.FNCを登録した状態でX-BASICを起動し，リスト1を入力してください(CARD.FNCでも大丈夫です)。RUNしてそのまま遊ぶのもよし，Cコンパイラをお持ちの方はコンパイルしてから遊ぶのもよいでしょう。速度的には，コンパイルしなくても十分遊べるものになっています。

ルール解説

簡単にルールを解説します。

使用するカードは各スートのA～10までの40枚。このゲームは2～5人くらいで遊ぶのが適当でしょう。今回のプログラムではプレイヤーを4人としています。

最初に親を決め，親は5（人数＋1）枚のカードを場にさらします。5枚目は親の台札として，残りの4枚のうち1枚ずつ子が自分の台札を選んでいきます。ここで賭金(マッチ棒)を決め，2枚まで新たな札を台札に加えて手を作っていきます。役は2枚または3枚のカードで構成していきます。

このようにして手札を作り終わると，子は親と役を比べ，賭金をやり取りして1ゲーム終了です。このゲームでは常に親とのみ賭けを行い，子同士では賭金をやり取りしません。

前回いちばん強い役を作った人が親になりゲームは続けられます。誰かのマッチ棒がなくなったらゲームは終了です。

このゲームのつらいのは親のときです。子が3人とも5本ずつ賭けてきたときにブタでも出そうものなら一気に破産してしまいかねません。

役の解説

では，肝心の役を解説しましょう。役の種類は大きく分けて，親にだけ可能なもの，子にだけ可能なもの，両者でともに可能なものに分かれます。今回採用したものは以下のとおりです。

●親だけ

シッピン（Aと4の組み合わせ）

クッピン（Aと9の組み合わせ）

●子だけ

アラシ（同じ数が3枚）

●共通

ナラビ（1，2，3枚が連続する数）

ハサミ（1枚目と3枚目が同じ数）

カブ（カードの合計の下1桁が9）

ブタ（カードの合計の下1桁が0）

シッピンとクッピンは手札2枚のときのみ有効な役で，札の順序はどうでもかまいません。

さて，これらの役とカードの合計の下1桁の数値による役の強さは，

アラシ＞ナラビ＞クッピン＞シッピン＞ハサミ＞カブ＞8＞7＞……1＞ブタ

となっています。

例を挙げてみると，親の場合，

2，3，4　：ナラビ
9，A　：クッピン
A，4　：シッピン
10，6，10　：ハサミ
9，4，6　：カブ
1，4，5　：ブタ
3，3，3　：ハサミ

となります。

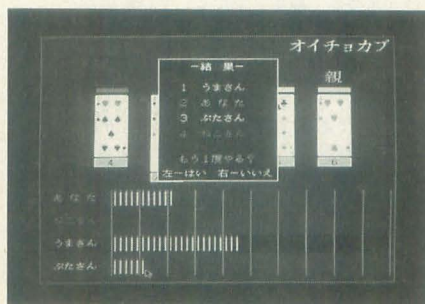
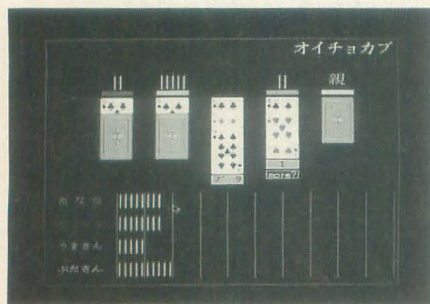
また，子の場合は，

2，3，4　：カブ
9，A　：ブタ
A，4　：5
10，6，10　：6
9，4，6　：カブ
1，4，5　：ブタ
3，3，3　：アラシ

です。なお，親・子いずれの場合とも，

9，10，A　：ブタ

です。10とAはつながりません。

実は今回採用した以外にもいろいろと役があり，細かいルールは地方によっても差があります。その点はあらかじめご了承ください。

ゲーム中は常に評価された役が表示されていますので，よくわからない人でもしばらくプレイしてみればコツがわかると思います。とりあえず起動してみるのがいちばんでしょう。

操作法

操作はすべてマウスで行います。自分の台札の選択と必要な操作は，台札に新しい札を加えるか？　の選択と賭金の選択のみ(子の場合)，親の場合は台札と賭金の選択もありません。

新しい札を加えるときは左クリック，いらないときは右クリックしてください。賭金はマッチ棒の部分を左クリックし，枠を引っ張って本数を指定します。右クリックはキャンセルです。最大5本まで賭けることができます。

その他，ゲームの進行上必要なときには左クリックをしてください。

＊　＊　＊

このままでは雰囲気が出ないという方は1月号の付録ディスクに収録されていたカードエディタを使って花札のカードを自分でデザインしてみるのもよいでしょう。CARDDRVやCARD2.FNCはなにもトランプゲーム専用のものではないのですから。

特にCARD.FNCからCARD2.FNCにバージョンアップされたときには，あらゆるカードゲームに対応することをテーマに256色データへの対応やカード枚数制限の緩和 (CARD2.FNCでは200枚まで)，コンパイラへの対応などが図られました(「MON○TER ○AKERシリーズを完全に再現する」ための仕様変更だったような……)。

ただし花札の場合，それを使ったゲームが少ないのであまり使い回しがきかないというのが難点ですが……。

表1　変数表

変数	内容
exc	親の1枚目のカードの内容
basyo	どの場所に賭けたか？
oya	親は誰か？
zero	マッチの本数がゼロ？
fin	ゲーム終了？
card( )	カード40枚の内容
hand( )	手札の内容
po( )	役の強さ
ho( )	何本賭けたか？
wh( )	誰が賭けたか？
ma( )	マッチのグラフィックデータ
hide( )	2枚目に配ったカードの内容
cy( )	役を表示するY座標
so( )	ソート用
co( )	カラーデータ
per( )	コンピュータが2枚目のカードを引く確率
jun( )	賭ける順番
la( )	順位
match( )	マッチの本数

リスト1

```
  10 /*
  20 /* The Super Kabu
  30 /*    Written by Azuron 1991 2.8(Fri)
  40 /*
  50 int i,j,k,x,y,n,bl,br,exc,basyo,oya,zero,fin
  60 dim int card(40),hand(15),po(5),ho(5),wh(5)
  70 dim int ma(128),hide(5),cy(5),so(4),co(4)={3,5,7,1
1}
  80 dim int match(4),jun(4),la(4),per(9)={0,0,0,0,5,10
,50,80,100}
  90 dim str nam(4)[8]={"あ な た","ねこさん","うまさん","ぶたさ
ん"}
 100 /*
 110 for i=0 to 4
 120   po(i)=i:wh(i)=i-1
 130 next
 140 /*
 150 music()
 160 jqk_cut()
 170 repeat
 180   prep()
 190   repeat
 200     junban()
 210     shuffle()
 220     init()
 230     ba_put()
 240     sbet3()
 250     deal()
 260     syobu()
 270     show()
 280     pay()
 290   until zero=1
 300   fin=result()
 310 until fin=0
 320 /*
 330 screen 2,0,1,1
 340 end
 350 /*
 360 func sbet3() /* 3人分まとめて選択&賭ける
 370   int i
 380   for i=0 to 2
 390     select(i)
 400     bet(i)
 410   next
 420 endfunc
 430 /*
 440 func ba_put() /* 場にカードを出す
 450   int i,x=80
 460   apage(2)
 470   for i=0 to 4
 480     m_play(1)
 490     hand(i*3)=conv(card(i))
 500     if i=4 and oya<>0 then {
 510       exc=card(i):ura(x,120)
 520         break
 530     }
 540     c_put(x,120,card(i))
 550     x=x+80
 560     mwait(1)
 570   next
 580 endfunc
 590 /*
 600 func select(w) /* 賭けるカードを選択
 610   int x=0,y,pl
 620   pl=jun(w)
 630   myself(w,1)
 640   apage(1)
 650   flbx(400,104,447,114,co(oya),co(oya)-1)
 660   msarea(80,100,383,195)
 670   setmspos(80,100)
 680   if pl<>0 then x=AI(1,0)
 690   repeat
 700     msstat(n,n,bl,br)
 710     mspos(x,y)
 720     x=((x-80)¥80)*80+80
 730     box(x-2,118,x+48,217,15,&HAAAA):wait(1)
 740     box(x-2,118,x+48,217,0):wait(1)
 750     basyo=(x-80)¥80
 760  until (bl+br<>0 and wh(basyo)=-1) or pl<>0
 770  flbx(x,104,x+47,114,co(pl),co(pl)-1)
 780  wh(basyo)=pl
 790  wh(4)=oya
 800  wait(2)
 810 endfunc
 820 /*
 830 func bet(w) /* 賭ける
 840   int x,y,mx,my,xx,h,pl
 850   pl=jun(w)
 860   apage(0)
 870   repeat
 880     msarea(0,0,511,511)
 890     y=pl*48+316
 900     setmspos(104,y)
 910     x=match(pl)*8+104
 920     msarea(104,y,x,y+33)
 930     wait(2)
 940     if pl<>0 then h=AI(2,w):break
 950     repeat
 960       msstat(n,n,bl,br)
 970       mspos(mx,my)
 980     until bl+br<>0
 990     xx=(mx¥8)*8
1000     if x>xx+39 then x=xx+39
1010     msarea(xx,y,x+1,y+33)
1020     x=xx
1030     wait(2)
1040     repeat
1050       msstat(n,n,bl,br)
```

```
1060        mspos(mx,my)
1070        mx=(mx¥8)*8
1080        box(x,y,mx,y+33,15,&HAAAA)
1090        box(x,y,mx,y+33,0)
1100      until bl+br<>0
1110      h=(mx-x)¥8
1120    until bl=-1 and h<>0
1130    ho(basyo)=h
1140    ma_kesu1(pl,h)
1150    ma_disp2(h)
1160    myself(w+1,0)
1170    wait(3)
1180  endfunc
1190  /*
1200  func deal() /* カードを配る
1210    int i,x=0,y=0,z,d,st=5
1220    for i=0 to 4
1230      apage(2)
1240      x=x+80:y=152
1250      d=wh(i)
1260      if i=4 and oya<>0 then c_put(x,120,exc)
1270      m_play(1)
1280      if d<1 then c_put(x,y,card(st)) else ura(x,y)
1290      hide(i)=card(st)
1300      hand(i*3+1)=conv(card(st))
1310      cy(i)=y
1320      if d<1 then yaku_disp(i,y)
1330      st=st+1
1340      mwait(1)
1350      if d=0 then z=more(x) else z=AI(3,i)
1360      if z=0 then continue
1370      y=y+32
1380      apage(1)
1390      m_play(1)
1400      c_put(x,y,card(st))
1410      cy(i)=y
1420      hand(i*3+2)=conv(card(st))
1430      if d<1 then yaku_disp(i,y)
1440      st=st+1
1450      mwait(1)
1460    next
1470  endfunc
1480  /*
1490  func pay() /* 支払い
1500    int i,p,h,w
1510    for i=0 to 3
1520      p=po(i):h=ho(i):w=wh(i)
1530      if w=-1 then continue
1540      if po(4)>p then {
1550        ma_kesu2(i)
1560        ma_disp1(oya,h)
1570       }
1580       if po(4)<p then {
1590       ma_kesu1(oya,h)
1600       ma_disp1(w,h*2)
1610       }
1620      if po(4)=p then {
1630        ma_disp1(w,h)
1640        ma_kesu2(i)
1650      }
1660    next
1670    for i=0 to 3
1680      if match(i)=0 then zero=1
1690    next
1700    click()
1710  endfunc
1720  /*
1730  func sort() /* ソート
1740    int i,j,k,a,b,c
1750    for i=1 to 4
1760      k=3
1770      for j=1 to 4
1780        a=so(k):b=so(k+1)
1790        if b>a then {
1800          so(k)=b:so(k+1)=a
1810          c=la(k):la(k)=la(k+1):la(k+1)=c
1820        }
1830        k=k-1
1840      next
1850    next
1860  endfunc
1870  /*
1880  func junban() /* 順番を決める
1890    int i,j=4,c=0
1900    for i=0 to 4
1910      if wh(j)=-1 then j=j-1:continue
1920      so(c)=po(j):la(c)=wh(j)
1930      c=c+1:j=j-1
1940    next
1950    sort()
1960    jun(3)=la(0)
1970    for i=0 to 2:jun(i)=la(i+1):next
1980    you_oya(0)
1990    oya=jun(3)
2000    you_oya(1)
```

```
2010  endfunc
2020  /*
2030  func show() /* カードをオープン
2040    int i,x=0
2050    for i=0 to 4
2060      x=x+80
2070      if wh(i)<1 then continue
2080      yaku_disp(i,cy(i))
2090      m_play(1)
2100      c_put(x,152,hide(i))
2110      mwait(1)
2120    next
2130  endfunc
2140  /*
2150  func syobu() /* 勝負! 表示
2160    apage(0)
2170    flbx(204,220,308,252,12,13)
2180    sym(213,224,"勝 負!",2,15)
2190    click()
2200    fill(204,220,308,252,0)
2210  endfunc
2220  /*
2230  func yaku_disp(w,z) /* 役表示
2240    str ya[5]
2250    x=(w)*80+80:y=z+98
2260    k=yaku_judge(w)
2270    ya=itoa(k)      /*12345
2280    if k= 0 then ya="ﾌﾞ  ﾀ"
2290    if k= 9 then ya="ｶ  ﾌﾞ"
2300    if k=10 then ya="ﾊｻﾐ"
2310    if k=11 then ya="ｼｯﾋﾟﾝ"
2320    if k=12 then ya="ｸｯﾋﾟﾝ"
2330    if k=13 then ya="ﾅﾗﾋﾞ"
2340    if k=14 then ya="ｱﾗｼ"
2350    apage(2)
2360    flbx(x,y,x+46,y+17,14,15)
2370    sym((x+24)-(len(ya)*8¥2),y+1,ya,1,4)
2380  endfunc
2390  /*
2400  func more(x) /* もっと?
2410    flbx(x,272,x+46,288,12,13)
2420    sym(x+4,273,"more?",1,15)
2430    click()
2440    fill(x,272,x+46,288,0)
2450    wait(2)
2460    return(bl)
2470  endfunc
2480  /*
2490  func AI(fu,w) /* 人工知能
2500    int z,xx,bx,h
2510    if fu=1 then {
2520      repeat
2530        x=int((rand()/32768.0)*4.0)
2540      until wh(x)=-1
2550      x=x*80+80
2560      for i=80 to x:setmspos(i,100):i=i+9:next
2570      return(x)
2580    }
2590    if fu=2 then {
2600      y=jun(w)*48+316
2610      repeat
2620        h=int((rand()/32768.0)*5.0+1)
2630      until match(jun(w))-h>-1
2640      xx=h*8+104
2650      for i=104 to xx
2660        setmspos(xx,y)
2670        bx=(i¥8)*8
2680        box(104,y,bx,y+33,15,&HAAAA)
2690        box(104,y,bx,y+33,0)
2700      next
2710      wait(3)
2720    return(h)
2730    }
2740    if fu=3 then {
2750      z=yaku_judge(w)
2760      if z>8 then return(0)
2770      z=int((rand()/32768.0)*per(z))
2780      if z=0 then return(-1)
2790      return(0)
2800    }
2810  endfunc
2820  /*
2830  func yaku_judge(w) /* 役を判定
2840    int z,h1,h2,h3,ju=0
2850    z=w*3
2860    h1=hand(z):h2=hand(z+1):h3=hand(z+2)
2870    while w=4 and h3=0
2880      if h1<>1 and h2<>1 then break
2890      if abs(h1-h2)=3 then ju=11
2900      if abs(h1-h2)=8 then ju=12
2910      break
2920    endwhile
2930    if ju<>0 then po(w)=ju:return(ju)
2940    if h1=h3 then ju=10
2950    if h1+1=h2 and h1+2=h3 then ju=13
```

▶2月某日，M大の入試のあと。ふとキ○ラヤの店頭を見るとカラフルな箱が積んである。スーパーファミコンだ。半年先まで予約でいっぱいなんじゃないのかと思いつつ，あと15,000円でどうやって今月切り抜けるのか考え込む私であった。　富田　祐樹(18)東京都

```
2960    if h1=h2 and h1=h3 and w<4 then ju=14
2970    if ju<>0 then po(w)=ju:return(ju)
2980    ju=h1+h2+h3
2990    ju=ju-((ju¥10)*10)
3000    po(w)=ju
3010    return(ju)
3020 endfunc
3030 /*
3040 func result() /* 結果
3050   int y=96
3060   for i=0 to 3
3070     so(i)=match(i):la(i)=i
3080   next
3090   sort()
3100   apage(0)
3110   flbx(170,44,342,298,1,15)
3120   sym(216,52,"-結 果-",1,15)
3130   for i=0 to 3
3140     sym(200,y,chr$(&H82)+chr$(&H50+i)+"  "+nam(la(
i)),1,co(la(i)))
3150     y=y+32
3160   next
3170   sym(200,244,"もう1度やる?",1,9)
3180   sym(177,274,"左-はい 右-いいえ",1,13)
3190   click()
3200   return(bl)
3210 endfunc
3220 /*
3230 func ma_disp1(w,hw) /* マッチ表示1
3240   int i,x,y
3250   apage(0)
3260   x=match(w)*8+105
3270   y=w*48+317
3280   for i=1 to hw
3290     m_play(2):put(x,y,x+7,y+31,ma)
3300     x=x+8
3310     mwait(2)
3320   next
3330   match(w)=match(w)+hw
3340 endfunc
3350 /*
3360 func ma_disp2(hw) /* マッチ表示2
3370   apage(0)
3380   x=(24-(hw*4)+80)+80*basyo
3390   for i=1 to hw
3400     put(x,70,x+7,101,ma)
3410     x=x+8
3420   next
3430 endfunc
3440 /*
3450 func ma_kesu1(w,hw) /* マッチを消す1
3460   apage(0)
3470   match(w)=match(w)-hw
3480   x=match(w)*8+105
3490   y=w*48+317
3500   for i=1 to hw
3510     m_play(2):fill(x,y,x+7,y+31,0)
3520     mwait(2)
3530     x=x+8
3540   next
3550   if match(w)<0 then match(w)=0
3560 endfunc
3570 /*
3580 func ma_kesu2(w) /* マッチを消す2
3590   apage(0)
3600   x=w*80+80
3610   fill(x,70,x+80,101,0)
3620 endfunc
3630 /*
3640 func shuffle() /* シャッフル
3650   int i,a,b,k
3660   for i=0 to 99
3670     a=int((rand()/32768.0)*40):b=int((rand()/32768
.0)*40)
3680     k=card(a):card(a)=card(b):card(b)=k
3690   next
3700 endfunc
3710 /*
3720 func jqk_cut() /* J,Q,Kを抜く
3730   int i,c=0,d=0
3740   dim int jqk(9)={11,12,13,24,25,26,37,38,39}
3750   for i=1 to 49
3760     if i=jqk(c) then c=c+1:continue
3770     card(d)=i
3780     d=d+1
3790   next
3800 endfunc
3810 /*
3820 func you_oya(sw) /* 親にバー表示
3830   apage(2)
3840   y=oya*48+317
3850   fill(4,y-2,504,y+34,sw)
3860 endfunc
3870 /*
3880 func myself(w,sw) /* バー表示
3890   int y,yy
3900   apage(2)
3910   y=jun(w)*48+324
3920   if w<>0 then {
3930     yy=jun(w-1)*48+324
3940     fill(12,yy,92,yy+15,0)
3950   }
3960   if sw=1 then fill(12,y,92,y+15,15)
3970 endfunc
3980 /*
3990 func sym(x,y,s;str,z,c) /* シンボリック
4000   symbol(x,y,s,1,1,z,c,0)
4010 endfunc
4020 /*
4030 func flbx(x0,y0,x1,y1,c0,c1) /* フィル&ボックス
4040   fill(x0,y0,x1,y1,c0)
4050   box(x0,y0,x1,y1,c1)
4060 endfunc
4070 /*
4080 func click() /* クリックされるまで待つ
4090   repeat:msstat(n,n,bl,br):until bl+br<>0
4100 endfunc
4110 /*
4120 func mwait(ch) /* ウエイト
4130   repeat:until m_stat(ch)=0
4140 endfunc
4150 /*
4160 func ura(x,y) /* カードを裏返しに置く
4170   c_put(x,y,0)
4180 endfunc
4190 /*
4200 func conv(v) /* コンバート
4210   return(v-(v¥13)*13)
4220 endfunc
4230 /*
4240 func wait(ch;int) /* ウエイト
4250   m_play(ch+2)
4260   repeat:until m_stat(ch+2)=0
4270 endfunc
4280 /*
4290 func init() /* 変数初期化
4300   zero=0:fin=1
4310   for i=0 to 3:wh(i)=-1:next
4320   for i=0 to 15:hand(i)=0:next
4330   for i=0 to 2
4340     apage(i):fill(0,20,511,314,0)
4350   next
4360 endfunc
4370 /*
4380 func music() /* 音楽
4390   m_init()
4400   for i=1 to 6
4410     m_alloc(i,500):m_assign(i,i)
4420   next
4430   m_tempo(200)
4440   m_trk(1,"@59v15c8")
4450   m_trk(2,"q1@15v13o4c64")
4460   m_trk(3,"r64.")
4470   m_trk(4,"r8")
4480   m_trk(5,"r2")
4490   m_trk(6,"r1")
4500 endfunc
4510 /*
4520 func prep() /* 準備
4530   str kabu[12]="オイチョカブ"
4540   srand(val(mid$(time$,4,2)+right$(time$,2)))
4550   screen 1,1,1,1
4560   console ,,0
4570   vpage(15)
4580   apage(3)
4590   fill(3,0,4,31,13)
4600   fill(3,0,4,3,5)
4610   get(0,0,7,31,ma)
4620   flbx(0,0,511.511,8,9)
4630   sym(362,8,kabu,2,0)
4640   sym(360,6,kabu,2,15)
4650   sym(413,72,"親",2,0)
4660   sym(411,70,"親",2,13)
4670   apage(1)
4680   for i=0 to 3
4690     sym(20,i*48+324,nam(i),1,co(i))
4700   next
4710   for x=104 to 510
4720     line(x,317,x,496,14)
4730     x=x+39
4740   next
4750   for i=0 to 3
4760     match(i)=0:ma_disp1(i,10)
4770   next
4780   mouse(4):mouse(1)
4790 endfunc
```

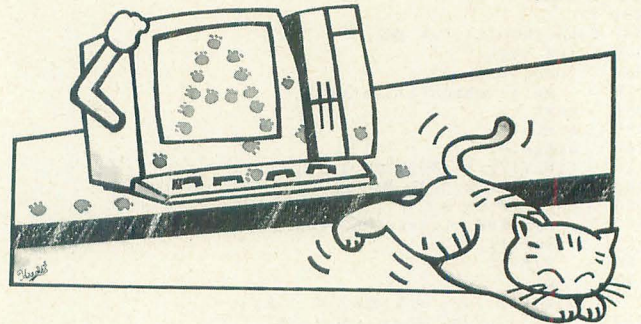
▶ひとり暮らしを始めて，もう1年がたちました。最近はあちこちのスーパーを回って，1円でも安いものを買うことが楽しく感じられるようになってしまいました。こりゃ，将来は主夫にでもなるしかないな，と思う今日この頃です。　松永　正弘(20)京都府

新連載

吾輩はX68000である
[第1回]

まずAより表示せよ

Izumi Daisuke 泉 大介

パソコンにはユーザーという主がいる
マシンにもそれなりに都合というものがある
そこには不思議議な関係が………

吾輩はX68000である。名前はまだない。世のなかには自分のパソコンに名前をつけるという趣味を持つ輩が存在し，吾輩の主人たる泉大介氏の知り合いのなかにも自分のマシンを「ともよちゃん」と呼んでいる輩が在る。付けられる名前が決まって女性名なのは，かような趣味を持つ御仁の知り合いが男性ばかりだからなのであろう。とすれば，女性のなかには自分のマシンを「とおるくん」などと呼ぶ人が在るのだろうか。吾輩が「ゆきちゃん」とか「あきほちゃん」などと呼ばれでもした日には，怒りのあまりキャリングハンドルで天を突くところであるが，幸いにも御仁にはこのような趣味はない。御仁のところへやってきて，はや，4年が過ぎようとしているが，あいかわらず人に紹介するときには「うちのロクハチ」で済ましている。

吾輩の誕生

吾輩は4年前，当時最高のスペックを誇るパーソナルコンピュータとして製を享けた。吾輩の頭脳にはCPU界の良心と賛されたMC68000が採用され，グラフィック表示能力は高解像度の6万5千色以上にも及ぶ。サウンドはステレオのFM音源8声＋人の声も再生できるADPCM。文字は8ドット，12ドット，16ドット，24ドットの4段構えである。すべてが当時のパソコンが持つスペックを凌駕しており，吾輩はパーソナルワークステーションと命名された。

御仁は，荻野目洋子の実写と見紛うばかりのグラフィックを表示しながら彼女の歌を歌う吾輩の先輩に魅せられ，吾輩の購入を決意したのだと聞いている。市ヶ谷の某所で行われた先輩のデモを見た御仁はあまりの能力に驚き，「ひでぇ」と漏らしたということだ。これではほかのパソコンの立つ瀬がないじゃないか，という意味での言葉だったらしいが，先輩の能力に言葉を失った人々の間を流れる沈黙を破ったこのひと言はかなりインパクトがあったらしい。Oh!MZに掲載された後日談でもこの言葉が伝えられることとなった。

現在では吾輩のことを「ワークステーション」であると考える人々は少ないようだ。逆に，ビジネスパソコンとして御仁が蔑視する98某のほうがCPUに20MHzの80386を搭載してその能力を高め，ワークステーションと変わらぬ能力をもつパソコンとして勇名を馳せている。さらには80486を搭載したモデルまで登場し，その処理能力はまさにワークステーションと呼ぶにたるレベルに達してきた。

しかしながら，そのマシンの上で行われていることは単なる事務処理の縮小再生産に過ぎないのではないだろうか。3百万台を超える販売実績を持つとされる98某が行ってきたことは，オフィスの仕事を家庭に持ち込むことだけである。10MHzのMC68000は確かに処理能力的には98某に及ばないかもしれないが，パソコンはCPU能力という物差しのみで測られるものにあらず。マシンスペックのすべてを動員して作成されるアクション系のゲームは，そのマシンの総合力を判断するひとつの基準となるべきものであるが，この分野においては吾輩はいまだに他の追従を許さず，ゲームフリークの憧れるパソコンのトップに君臨していると自負しておる。

吾輩はいかに文字を表示するか

吾輩の特長のひとつとして，テキスト画面がビットマップであることが挙げられる。テキスト画面とは文字を表示する画面のことである。我々パソコンがなぜ画面に文字を表示できるのか疑問に思われたことはないだろうか。テレビでも使っているブラウン管につながっているから？　外見上は確かにそのとおりだが，内部ではいささか複雑な処理が行われている。テレビ受像機が絵を表示できるのは，元となる信号を電波として受け取っているからだが，吾輩がディスプレイに文字や絵を表示するためにもこれと同じような信号を内部で作ってやらなければならない。テキスト画面はこのために用意されたものである。

ディスプレイに表示される文字をよくよく眺めてみる

と，曲線部や斜線部にガタガタがあるのがわかる。これは我々が表示する文字は小さな点を集めて作られているからで，たとえばAという文字は図1-aのように点を集めて作ってある。確認したければ，吾輩が携えたX-BASICで実験してみるのがよかろう。

X-BASICが起動したなら，まず，

```
screen 2,0,1,1
```

とキーボードから打ち込みリターンキーを押す。これで画面が一瞬暗転するはずである。続いて，

```
symbol(160,160,"A",16,16,15,0)
```

と打ち込みリターンキーを押すと，Aという文字が拡大して表示される。図1-aと比較されたい。

ほかの文字を拡大表示したければ，

```
wipe( )
```

と入力してリターンキーを押す。これで拡大表示したAの文字が画面から消えるはずである。続いて上の“A”の部分を変更して試せばいい。変更はカーソルキーを使って画面で点滅しているカーソルを“A”のところまで移動させ，表示させたい文字を入力してリターンキーを押すだけである。wipe( )を怠ると，拡大表示されたAの上に次の文字が重なって表示され，訳のわからぬ模様が表示されるので注意されたい。

screenやsymbolが何を行っているのかは，いずれ詳しく話す機会もあろう。今回は点が文字を構成していることを確認するだけにとどめておくことにする。実験が終わってX-BASICを終了するには，

```
system
```

と入力してリターンキーを押せばいい。再びビジュアルシェルあるいはSX-WINDOWに戻ることができる。

文字が点の集合で表現されていることはこれでおわかりかと思うが，実際に吾輩がどのようにこれを扱っているのかを次にお話ししよう。吾輩はコンピュータである。ご存じのようにコンピュータは電圧の高・低をもとにして動作している。これを1，0で表現することにすれば，吾輩は0，1の2つの記号をもとに動作していると言い換えることができる。これが，我々コンピュータが0と1しかない世界，すなわち，2進数の世界で動作するといわれる所以である。文字を表す点の集合を，点があるところを1，点がないところを0で表現すると，Aという文字は図1-bのようになる。これこそが吾輩が実際に文字を扱うときの形態なのである。

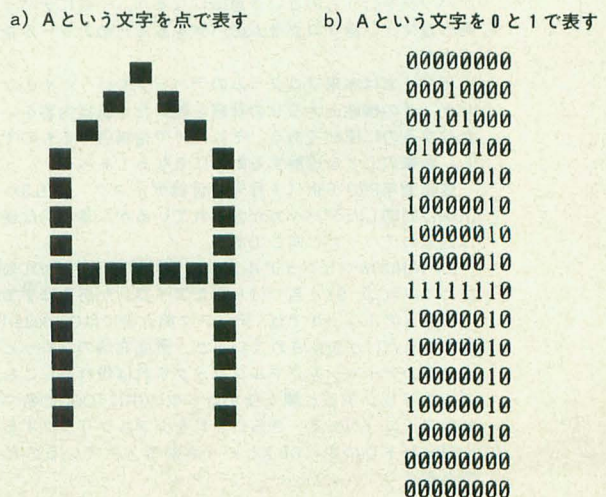
1行あたり0，1が8個詰め込んであることに疑問をお持ちの諸兄が在るかもしれない。これは，8個の0，1，すなわち8桁の2進数が吾輩にとって基本ともいえるデータの形で，苦労なく扱える単位だからである。同時に，最右桁が0，つまり点を表示しない状態になっていることにより，次の文字とくっついてしまうことも防止されている。同様にAという文字の上には0ばかりからなる空きがひとつ，下には空きが2つ用意されているが，これは上下に並んだ文字がくっついてしまうことを防止している。点のひとつをドットというが，(表示しない点も考慮に入れれば）図1-bはAという文字が横方向に8つの点，縦方向に16個の点を並べて作られていることを意味している。8×16ドットフォントと呼ばれるのはこのためだ。

実際の表示は，吾輩がこのデータをテキストVRAM（ヴイラム）と呼ばれている特別なメモリに複写することによって行われる。テキストVRAMはデータの1，0に応じてドットを点・消燈させる信号を作り出す特別なメモリである。この信号がディスプレイに送られ，画面上に文字が表示されることになる。テキストVRAMに複写するデータに制限はないから，自分の表示したい図柄を0，1の集合で表現しさえすれば，それをテキストVRAMに複写することによってどんな文字でも表示することが可能となるのである。先にテキスト画面と呼んだのは，このテキストVRAMにほかならない。

御仁の悪戯

御仁はデバッガを起動して，なにやら悪戯を始めようとしている。デバッガとはマシン語プログラムの動作を確かめる（デバッグする）ために作成されたプログラムで，XC（C compiler PRO-68K）などを購入すると付属してくるものである。

何をするのかと見ていれば，なんとテキストVRAMに自分でデータをセットするつもりらしい。デバッガにはメモリにデータをセットする機能が備わっているから，これを使って自分で文字を描こうというのである。メモリには0から順番に番号が振られている。広大なメモリ内の特定の場所を示すという意味で，この番号は「番地」あるいは「アドレス」という名前で呼ばれている。そしてひとつのアドレスには8桁の2進数を格納しておくこ

図1　点を集めてAを作る

とができる。先に吾輩にとって扱いやすいデータ単位として8桁の2進数を挙げたが，扱いやすいというのはこういうわけだったのである。

テキストVRAMはE00000$_H$番地から始まっている。最後に$_H$がついているのは，E00000というのが16進数であることを意味している。2進数だの16進数だのと面倒臭そうだが，これは吾輩のせいではない。8桁の2進数を16進数なら2桁で表現できるため，簡略で便利だと人間が勝手に使い始めたのである。念のため，2進数・16進数・10進数の対応表を，表1に用意しておいたので参照されたい。テキストVRAMのアドレスは6桁の16進数で表現されているが，これを2進数で表現すると24桁に及ぶ。1と0を24個も羅列する勇気はさすがになかったに違いない。

前述のとおりE00000$_H$番地には8桁の2進数を格納することができ，この最左桁が画面左上のドットに相当する。つまりE00000$_H$番地に10000000$_B$（$_B$は2進数の意味）というデータをセットすれば，画面左上の1ドットが点燈するわけである。11000000$_B$ならば画面左上の2ドットが，10100000$_B$ならば，1ドット空けて2つのドットが点燈することになる。「フムフム，なるほど」と御仁はしごく御満悦の様子。チェックが終わっていよいよ文字の表示にとりかかるらしい。

●メモリにデータをセットする

デバッガでメモリにデータをセットするには「me」コマンドを使用する。これはメモリエディット（Memory Edit：メモリ編集）の略である（たぶん）。デバッガを起動すると行頭の「－」に続いてカーソルが点滅しているであろう。ここで，

　－mes e00000

と入力してリターンキーを押せば，メモリのE00000$_H$番地にデータを書き込めるようになる。「me」の後ろに「s」がついているのは，8ビット（1バイト）ずつデータを書き込むことを指示している。最後に「w」をつければ2バイト（1ワード）ずつ，「l」をつければ4バイト（1ロングワード）ずつデータを書き込むこともできる。ちなみにwは省略可能なので，単に「me」とすれば，2バイト単位でデータをセットすることになる。画面には，

　00E00000　00：■

と表示されているはずである（■はカーソル）。これはE00000$_H$番地に，現在00$_H$というデータが書き込まれていることを意味している。つまり00000000$_B$，なにも表示しないというデータがセットされているわけだな。ここに100000000$_B$というデータをセットするなら，

　00E00000　00：_100000000■

と書き込んでリターンキーを押せばいい。最初の「_」は数字が2進数であることを意味している。これで画面の左上に点が表示されたはずである。画面は続いて，

　00E00000　00：_100000000
　00E00001　00：■

となる。次はE00001$_H$番地にデータをセットするのだというわけだ。

さて，E00000$_H$番地は画面の左上の8ドットだったが，E00001$_H$番地は画面のどこにあたるのだろう。大方の予想どおり，その右隣の8ドットがこれにあたる。よって，

　00E00000　00：_11111111
　00E00001　00：_11111111
　　　⋮

とデータをセットしていけば，画面の最上行にラインが引かれていくことになる。

現れたラインが水色をしているのでびっくりしたかもしれない。実は吾輩のテキストVRAMは図2のような構造をしている。いまデータをセットしたE00000$_H$番地というのはテキストプレーン0に相当し，ここにセットし

表1　2進数・16進数・10進数の対応表

2進数	16進数	10進数
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

16進数は覚えるしかないが，2進数は0，1，2，4，8だけを覚えておけば，あとは足し算で作ることができる。つまり，9は1000$_B$＋0001$_B$（8＋1）

デバッガとはなにか

プログラムの誤りをバグというのは周知のとおりである。そして，バグを修正することをデバッグという。デバッガはデバッグを行うものという意味になるが，一般にデバッガと呼べばマシン語プログラムのバグを取るためのツールを指すようだ。

デバッガは本来プログラムのデバッグを行うためのツールだが，その機能上メモリの状態を覗いたり直接内容をいじったりするのに便利である。それだけで結構遊べるものであるし，吾輩のことを理解する助けにもなるであろう。

謹賀新年PRO-68K（1月号の付録ディスク）にもSX-WINDOWに対応したデバッガが含まれているが，基本的な使い方は通常のデバッガと同じである。

SX-WINDOWやビジュアルシェルで付録ディスクのDISK4を眺めてみれば，SXと名づけられたアイコンがあるはずである（ビジュアルシェルでは，ディスク挿入時にはQuickstartディレクトリだけが表示されているが，画面右端のフロッピーディスクのアイコンをダブルクリックすれば現れる）。これをダブルクリックすると開くウィンドウの中にTOOLと名づけられたアイコンがある。さらにこれをダブルクリックすると開くウィンドウの中にDB.Xという名前で入っているのがそれである。

たデータは水色で表示されるのである。テキストプレーン1にセットすれば，それは黄色で表示される。では普段表示される白い文字はどうすれば表示できるのかというと，なんと，プレーン0とプレーン1に同じデータをセットしなければならない。つまり，

```
-mes e00000
00E00000    00:_11000000
00E00001    00:^C
-mes e20000
00E20000    00:_01100000
00E20001    00:^C
```

とすれば，画面左上に水色・白・黄色の3つの点が表示されるというわけである。なお,「^C」というのは, CTRLキーを押しながらCを押すことを意味している。この「^C」でメモリへのデータセットは中断される。もっとも，ディスプレイの個体差によっては3つの色を判別できないかもしれないが。

●御仁の失敗

「A」の文字を自分でセットしてみようとデバッガを使い始めた御仁は，最初の00000000_Bをセットし終わり，次の00010000_Bに取り掛かろうとしている。画面の2行目は図2からもおわかりのように，$E00080_H$番地である。吾輩のテキスト画面は横に1024ドットあり，ひとつの番地に8ドット分書き込めるのだから$1024\div 8=128=80_H$というわけである。「^C」でデータセットを中断して次は「mes e00080」でデータを書き込んでいく。3行目は$E00100_H$，4行目は$E00180_H$……，と進んでいくうちに悲劇は起きた。

ズリッ。画面がスクロールして，書きかけのAの文字はみごとに消えてしまったのである。さもありなん。デバッガで入力を続けていけば入力行が次第に画面下に移動していき，ついにはスクロールするのは自明の理である。テキスト画面の最上行に「A」の文字を書いていたのだから，スクロールに合わせその文字が消えてしまうもこれまた当然。吾輩のせいではない。ところが御仁はまだ諦めきれないらしく，恨めしそうな目で吾輩を見ている。吾輩のせいではないというに。

やがて御仁も気を取り直し，再び「A」の字の制作に取り掛かった。わざわざ自分で，しかも吾輩が持っている文字をセットしようというのも酔狂な話だが，転んでもただでは起きないところはさすがである。こうでなくてはプログラムなど作れるものではない。今度はテキストプレーン2を使って「A」の文字を描き始めた。テキストプレーンの2と3は吾輩のマウスカーソルを表示するのに使われておる。このため，画面がスクロールしてもその影響を受けない。ここならスクロールを気にせずに文字を表示できる。ちなみに，テキストプレーン2はマウスカーソルの下地の白に，テキストプレーン3はやや暗めの青に設定されている。

●テキスト画面をクリアする

御仁は，$E40000_H$，$E40080_H$，……，$E40700_H$，$E40780_H$とデータをセットし，ついに「A」の文字を完成した。ところがデバッガで入力した文字が邪魔で自分が作った「A」を見ることができない。いかにと見ていると，さすがに今度は躊躇もせず，

```
-f e00000 e3ffff 0
```

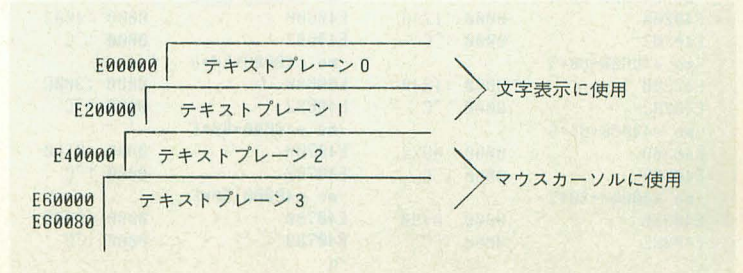
と入力した。「f」はメモリを特定のデータで埋め尽くすコマンドである。ここではテキストプレーン1の先頭である$E00000_H$から，テキストプレーン2の最後である$E3FFFF_H$までを0で埋め尽くしたわけである。これで画面上の文字はすべて消え，御仁の作成したマウスプレーンの文字だけが残ることになる。$E3FFFF_H$というのは，テキストプレーン3の先頭番地である$E40000_H$から1を引いた値である。

画面上の文字をすべて消したため，画面最下行に表示されていたファンクションキーも一緒に消えてしまったが，御仁は一向に気にせず自分の作成した文字を悦にいって眺めている。ではこの間に番地の計算方法を説明しておこう。

御仁はかつてはZ80のマシン語でならしたこともあり，16進数の計算を息でもするように難無くこなす（もっとも，さすがに16進数の掛け算は無理だ）が，こんな皆さんばかりではあるまい。そういった諸兄に吾輩から心ばかりのプレゼントがある。OPT.1キーを押しながらOPT.2キーを押してみていただきたい。画面に電卓が表示されたはずである。電卓窓の右端にはDECと表示されているが，これは現在10進数を扱っていることを意味している。ROLL UPキーを押すと，この表示はHEXに変わり16進数を扱えるようになる。ROLL DOWNキーを押すと再びDECに戻る。HEXにして次の計算を試していただきたい。

E00080+80

EはメインキーのEで入力し，0~9はキーボード右のテンキーで入力する。答えの表示はテンキーの「=」である。E00100と答えが表示されたことであろう。次にそのままENTERキーを押していただきたい。電卓で計算した答えはカーソル位置に複写される。これを使えば16進数など恐るるに足らず。たとえば文字の最後のライン，つまり縦16ドットめを書き込む番地は，

図2　テキストVRAMの構造

80×F＋E40000

で計算することができる。「×」はテンキーの「＊」で代用することになっている。計算途中でHEX・DECを切り替えることも可能なので，

80×（ROLL DOWN）15（ROLL UP）＋E40000

でもOKである。文字の１ラインめが80×0＋E40000なので，16ラインめは80×F＋E40000となるわけだ。テキストプレーン１の最後の番地，つまりテキストプレーン２の先頭番地の１つ手前がどうなるかをE40000－１で試してみていただきたい。また，128↔80_Hの変換も試されるとよかろう。

●デバッガの終了

最後にデバッガの終了方法である。終了は「q」コマンドで行うが，その前にテキストプレーン３に書いた文字を消しておかなければならない。これはテキストプレーン３を0で埋め尽くせばいい。つまり，

－f e40000 e5ffff 0

である。続いて「q」コマンドを実行すれば，自動的にビジュアルシェルやSX-WINDOWに復帰する。

図３　なにが現れるかはお楽しみ

a）お楽しみ１	b）お楽しみ２	c）お楽しみ３
0000	783C	00003F3F
0000	8442	00001092
0920	B29A	00001092
1240	B92A	00000848
1240	BC5A	1660E840
1240	BEAA	3360E420
0920	BEDA	3360E210
0490	BEAA	3360E210
2494	BEDA	66C1C210
4492	BEAA	66D9C108
8921	9ED2	66CDC108
8001	8EA2	66CDC084
4002	628C	CD99C484
300C	1830	CD998A42
0FF0	06C0	CD981242
0000	0100	599BBF3F

図４

```
-me e40000+80*0
E40000              0000 :0000
E40002              0000 :^C
-me e40000+80*1
E40080              0000 :0000
E40082              0000 :^C
-me e40000+80*2
E40100              0000 :0920
E40102              0000 :^C
-me e40000+80*3
E40180              0000 :1240
E40182              0000 :^C
-me e40000+80*4
E40200              0000 :1240
E40202              0000 :^C
-me e40000+80*5
E40280              0000 :1240
E40282              0000 :^C
-me e40000+80*6
E40300              0000 :0920
E40302              0000 :^C
-me e40000+80*7
E40380              0000 :0490
E40382              0000 :^C
-me e40000+80*8
E40400              0000 :2494
E40402              0000 :^C
-me e40000+80*9
E40480              0000 :4492
E40482              0000 :^C
-me e40000+80*A
E40500              0000 :8921
E40502              0000 :^C
-me e40000+80*B
E40580              0000 :8001
E40582              0000 :^C
-me e40000+80*C
E40600              0000 :4002
E40602              0000 :^C
-me e40000+80*D
E40680              0000 :300C
E40682              0000 :^C
-me e40000+80*E
E40700              0000 :0FF0
E40702              0000 :^C
-me e40000+80*F
E40780              0000 :0000
E40782              0000 :^C
-q
```

●おまけ

テキストVRAMに自分でデータをセットする楽しみを味わっていただこうと用意したのが図３である。なにぶんにも絵心がないもので，意図した形になっていないものもあるが入力して楽しんでいただきたい。データが２進数じゃないじゃないかって？　もちろんである。２進数にしてしまえば，入力せずとも絵がばれてしまう。吾輩はそんな愚は犯さない。

用意したデータは16進数になっている。デバッガではいきなり数字を書き並べたりすると16進数だと解釈されるので，

E40180　　0000：0920

のように入力すればいい。もちろん「mes」ではなく「me」コマンドである。図3-cはロングワードデータなので，こちらは「mel」を使う。図3-aをセットしていく様子を図４にまとめておいたので，作業の参考にしていただければと思う。

テキストVRAMいろいろ

吾輩のテキストVRAMはビットマップと呼ばれる方式であることを説明した。画面上のドットの点燈・消燈を，テキストVRAMに書き込む２進数の0と1で表現するわけである。しかしながら，ここのところの事情は，98某やX1シリーズなどでは若干異なっている。

我々が扱う文字には，それぞれ番号が与えてある。これはASCII（アスキー）コードと呼ばれるもので，たとえばAならば41_Hという番号になる。吾輩がドットの点燈・消燈を表す１，０の羅列をテキストVRAMに複写するのとは異なり，98某やX1シリーズなどはこのASCIIコードをテキストVRAMに複写するだけでいい。あとは自動的に文字の点燈・消燈データが取り出され，ディスプレイに送られるようになっている。ASCIIコードは８桁の２進数で表現できるため，吾輩がAの最初の８ドット分のデータを複写するのと同じ労力でひとつの文字を表示することが可能となる。なんと1/16の労力だ。昔々まだCPUの能力が低かった頃は，吾輩のような方法で文字を表示するのはあまりに荷が重かった。そのために考えられた方法なのである。

これによって確かに荷は軽くなったが，文字をイタリックにしたり太文字にしたりといった飾り付けを行うためには，専用のハードウェアが必要となってしまった。専用ハードウェアがなければそのマシンではイタリックも太文字も表示することができない。荷が軽くなった代償である。

吾輩はこのようなしがらみをきっぱりと取り払い，より自由度の高いマシンとして誕生したのであった。これもまた，パーソナルワークステーションと命名された所以であろう。

よいこのSX-WINDOW講座（第1回）

制御ボタンを使う

Nakamori Akira　中森　章

期待高まるSX-WINDOWですが，1月号で配布した資料を自力で理解できる人はまだまだ少ないことでしょう。そこで，具体的なプログラミング解説の始まりです。C言語を利用しますので，中森氏のもうひとつの連載も参考にしてください。

SX-WINDOWのドキュメントが1月号のおまけディスクで公開されて以来，すでに3カ月が過ぎようとしています。皆さんはSX-WINDOW上でのプログラムをたくさん作っていることと思います。えっ，まだそれほど作ってないのですか。おまけディスクのドキュメントが理解できないですって。ああよかった，実は私もそうなんです。あのドキュメントを理解するために日夜努力に励んでいますが，やっと最近になってプログラムの作り方が少しはわかるようになったところです。

というわけで，悪戦苦闘の結果を報告することがこの連載の目的です。きっと皆さんも私と同じようなところで悩み行き詰まっているのではないでしょうか。この連載ではSX-WINDOW上のアプリケーションを作りたいけど，どうしたらいいかわからないという人を対象としています。私といっしょにSX-WINDOWを勉強していきましょう。

まずは下準備

この連載ではプログラミングにC言語を用いることにします。アセンブラでガンガンやってもいいのですが，C言語で書いたほうがプログラムの流れを追いやすいと思います。そこで，まずC言語でプログラムを書くための下準備をしましょう。詳細は2月号の村田氏の記事[1)]を見てもらうことにして，とりあえずは，

　SXDEF.H
　SXLIB.H

がXCの標準ヘッダファイルのあるディレクトリに，

　SXLIB.A
　__MAINR.O

がXCの標準ライブラリと同じディレクトリにあることを確認してください（もしなければ，謹賀新年PRO-68Kの中からコピーしておいてください）。

その次はAR.Xを使って，

　AR－X SXLIB.A __MAIN.O

というコマンドを実行し，SXLIB.Aの中から__MAIN.Oを取り出します。これを，

　REN __MAIN.O __MAINC.O

というコマンドで__MAINC.Oなどとリネームしておいてください。この名前は何でもいいのですが，一応__MAINR.O(OBJR型のプログラムを作るためのスタートアップルーチン）に対応して__MAINC.O(OBJC型のプログラムを作るためのスタートアップルーチン）としておきます。そうそう，__MAIN.Oを取り出す操作はXCのver.2.0を使わない人（正確にはLK.Xのver.2.0を使わない人？）は不要です。これは，__MAIN.OをSXLIB.Aの中に入れておいたままだと，標準ライブラリCLIB.Aの__MAIN.Oとシンボルがぶつかって（早い話が2重定義）エラーになってしまうからです。

最後に，プログラムをコンパイルするためのバッチファイルを作ります。基本的には，コンパイルすべきプログラムとSXLIB.A（と__MAINC.O）をCC.Xに与えてやればよく，たとえば，リスト1に示すバッチファイルになります。使用するコンパイラに応じてリスト1の(a)～(c)のどれか1行をバッチファイルに書き込んでください。そのバッチファイルの名前を，たとえば，

　SXCC.BAT

としましょう。このとき，PROG.CというSX-WINDOW用のプログラムをコンパイルするためには，

　SXCC　PROG.C

というコマンドを実行します。もし，XCのver.1やGCCを使っていても，シンボルの2重定義エラーが出るようでしたらリスト1(b)を参考にして__MAINC.Oを先にリンクするようにバッチファイルを書き換えてください。

なお，XCを使用する場合，最適化オプションは指定しないでください。最適化を行うと予想外の動作（誤動作）をすることがあるようです。

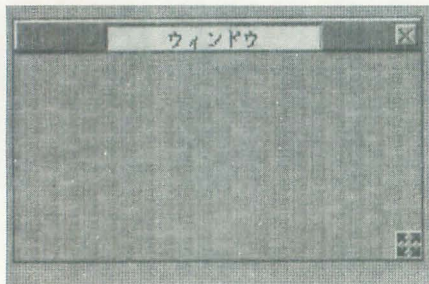
スケルトンを作る

SX-WINDOWのプログラムを作るための近道はスケルトン（骨格）プログラムを利用することです。これはSX-WINDOWの最低限の動きをするように作られているプログラムのことです。SX-WINDOWのプログラムはどれも構造が似通っているのでスケルトンプログラムを少し変更するだけで別のプログラムになってしまうのです。極端な話，スケルトンプログラムのアイドルイベントの処理とマウスの左ボタンの処理を書き換えるだけでプログラミングが終わってしまうこともあります。

スケルトンプログラムはネットに流れているものや2月号の村田氏のサンプルプログラムを利用することもできますが，自分専用のスケルトンプログラムを作っておくのもプログラムに愛着がわいていいものです。

リスト2に今回私の作成したスケルトンプログラムを示します。このプログラムは2月号の村田氏のサンプルプログラムが母

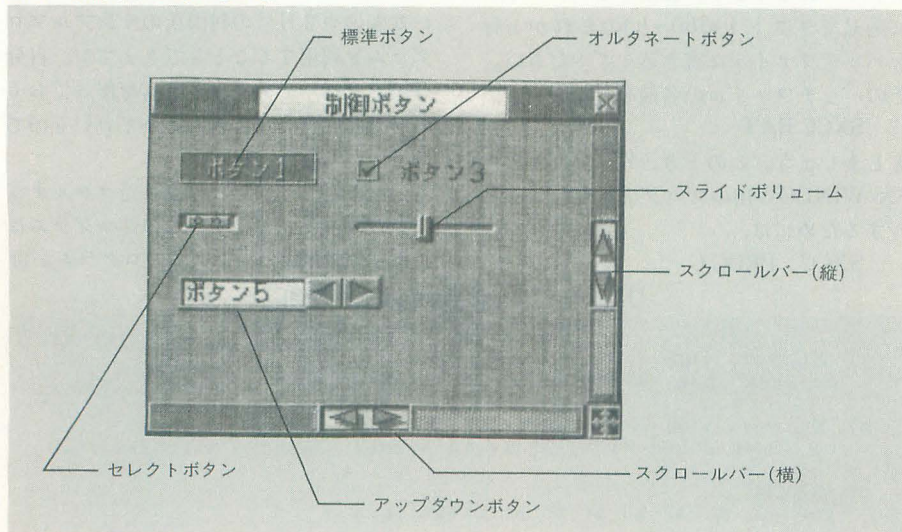
リスト1　コンパイル用バッチファイル

```
a) XC ver.1.0用
   cc /s4k /h8k  %1 %2 %3 %4 %5 %lib%¥sxlib.a

b) XC ver.1.0用
   cc /Gs4k /Gh8k %1 %2 %3 %4 %5 %lib%¥__mainc.o %lib%¥sxlib.a

c) GCC用
   gcc -O  %1 %2 %3 %4 %5 %lib%¥sxlib.a
```

スケルトンプログラム

体になっています。これから必要最小限の処理のみを取り出し，さらにポップアップメニューのための骨格を付け加えてみました。またポップアップメニューの追加は2月号の私の記事[2]が基になっています（あっちはアセンブリ言語でしたが)。詳しい説明が欲しい人は2月号の記事を参照してくださいね。

あと村田氏のプログラムからのちょっとひねくれた変更点（オリジナリティを持たせてみた）としては，イベントを表す構造体としてtseventの代わりにeventを使用したこと（構造体のフィールド名が違うだけ)，point_t型をlong型ではなくpoint型とlong型の共用体として宣言したことです。

制御ボタンとは

さてさて，やっとすべての下準備が終わりました。それでは連載第1回目のテーマである制御ボタンの話に入りましょう。制御ボタンとはSX-WINDOWの用語でコントロールと呼ばれている物体のことです。これは簡単にいうとウィンドウの飾りです。SX-WINDOW上のアプリケーションの中にはウィンドウ内にボタンがついたものやスクロールバーがついたものがあります。この飾りがコントロール，すなわち制御ボタンなのです。

SX-WINDOWでは7種類9つの制御ボタンが用意されています。具体的には，

- **●標準ボタン**
- **●セレクトボタン**
- **●オルタネートボタン**
- **●スライドボリューム**
- **●アップダウンボタン**
- **●スクロールバー（縦）**
- **●スクロールバー（横）**

の7種です。このうちスクロールバーには一般用とウィンドウ用があるのですが違いはよくわかりません。

言葉だけではそれらがどのようなものかわからないので実際にどんなものかお見せしましょう。図1がそれです。なかなか多種多彩なボタンが用意されていますね。今回は制御ボタンのうち，スクロールバー以外のものについて説明しましょう（スクロールバーを使用するときの処理がまだよくわからないもので)。

図1 制御ボタンのいろいろ

制御ボタンの操作

それでは実際に制御ボタンを扱うプログラムを書くことにしましょう。制御ボタンもその扱いはウィンドウやポップアップメニューと同様です。すなわち，

オープン
↓
操作
↓
クローズ

という流れで扱います。私たちが押さえておかなければならないのはそれぞれの具体的な処理です。順に説明しましょう。

1）制御ボタンのオープン

制御ボタン（コントロール）のオープンは，

CMOpen

という関数で行われます。関数の戻り値はハンドル（ポインタへのポインタ）と呼ばれるもので，以後オープンした制御ボタンに対する操作はこのハンドルを参照して行われます。CMOpenへの引数はおまけディスクのドキュメントを見てもらうことにして，ここでは引数に関する注意事項を述べましょう。

CMOpenの第3引数であるタイトルは制御ボタンの中に表示される文字列です。制御ボタンの種類によっては指定しても無意味なことがあります。このタイトルが意味を持つのは標準ボタン，オルタネートボタン，アップダウンボタンの3種類だけです。図1の制御ボタンに表示されている文字列はこのタイトルで指定したものです。なお，標準ボタン，オルタネートボタン，アップダウンボタンをオープンする場合，第2引数で指定する制御ボタンの大きさはタイトルが収まるだけの領域を持っていなければならないことはわかりますね。

CMOpenの第5，第6，第7引数は制御ボタンの値に関する引数です。一般的に制御ボタンはそれぞれ値というものを持っています。たとえば，オルタネートボタンはONのとき1，OFFのとき0という値を持っていますし，スライドボリュームなどはボリュームの位置に対応して変化する連続的な値を持っています。この値を知ることによって各制御ボタンがどのような状態にあるのかわかるようになっているのです。ただし，標準ボタンだけは値に意味がありません。標準ボタンはその場で押されたかどうかを問題とする制御ボタンなので，あとからその値を知ることに意味がないからです。

CMOpenの第8引数は制御ボタンの種類を示します。ここで7種類の制御ボタンを示すIDを指定するわけですが，値を16倍することを忘れないようにしましょう。

2）制御ボタンの操作

●左ボタンが押されたとき

制御ボタンはマウスの左ボタンを押して触りますから，制御ボタンに対する操作は通常マウス左ボタンダウンイベントの処理ルーチンに書かれます。マウスの左ボタンが押されたときの処理としてはウィンドウに対する移動などの処理もあります。制御ボタンに対する操作はウィンドウに対する操作をやったあとで行えばよいでしょう。

制御ボタンに対する操作は2通りの方法が考えられます。第1は堅実に細々とした操作を重ねていく方法，第2はコントロールマンに（ほとんど）すべてを任せる方法です。

制御ボタンの操作の基本は，

CMFind

関数と，

CMCheck

関数です。前者はマウスの指す位置（ローカル座標に変換しておかなければならない）に制御ボタンがあるかどうかを教えてくれる関数です。後者は制御ボタンのアニメーション処理をするための関数です。ここでいうアニメーション処理とは，押されたボタンを光らせるとかスライドボリュームのボリューム位置を変えるとか，オルタネートボタンにチェックマークをつけるとかいう処理を指します。ただし，CMCheck関数は制御ボタンの値を自動的に変えることは行いません（ただし，スライドボリュームの操作とスクロールバーで位置を指定してスクロールする場合の操作を除く）。

CMCheck関数は制御ボタンのどの部分が押されたかをパートコードとして返してきますから，必要な場合はパートコードに従ってコントロールの値を変更してやらなければなりません。たとえば，アップダウンボタンで右向き三角が押されたら値を1増やすとか，左向き三角が押されたら値を1減らすという処理が必要になります。このときに呼ばれる制御ボタンの現在値を得る関数は，

CMValueGet

で，制御ボタンに新しい値を設定する関数が，

CMValueSet

です。

以上のような処理を行うのが第1の堅実な方法です。これと比べると第2の方法はかなり簡単です。

SXCallCtrlM

という関数を呼ぶだけで第1の方法で行うほとんどすべての処理をやってくれます。押された制御ボタンがアップダウンボタンやスクロールバーでないなら，制御ボタンの値の変更まで行ってくれますからこんな便利な関数はありません。ただ，スクロールに関する処理を自動的に行うようになっている（ように見える）ので，スクロールバー以外を操作するときは余計な処理のために，第1の方法に比べて少しだけ実行時間がかかります。

SXCallCtrlMの第5引数（デスティネーションレクタングルへのポインタ）の意味がはっきりとはわかりませんがスクロールバーの処理を指定しない（第3引数と第4引数が0）なら意味がありません。図2に第1の方法と第2の方法を使う場合の操作手順をまとめておきます。

●アップデートイベント発生時

アップデートイベントが発生したらウィンドウ表示の書き直し処理とともに制御ボタンの描き直しをしなければなりません。あるウィンドウ内のすべての制御ボタンを描き直すための関数は，

CMDraw

関数です。この関数をウィンドウ表示を描き直したあとに呼び出します。

3) 制御ボタンのクローズ

制御ボタンのクローズのための関数は，

CMDispose

です。制御ボタンが不要になったらこの関数を呼び出して制御ボタンをウィンドウ上から消し去ります。

図2 制御ボタン（コントロール）の操作

サンプルプログラム

それでは制御ボタンを扱うサンプルプログラムを示しましょう。プログラムはリスト2のスケルトンプログラムを変更することで行います。リスト2では制御ボタンを追加することを前提として作成してあり，そのための，

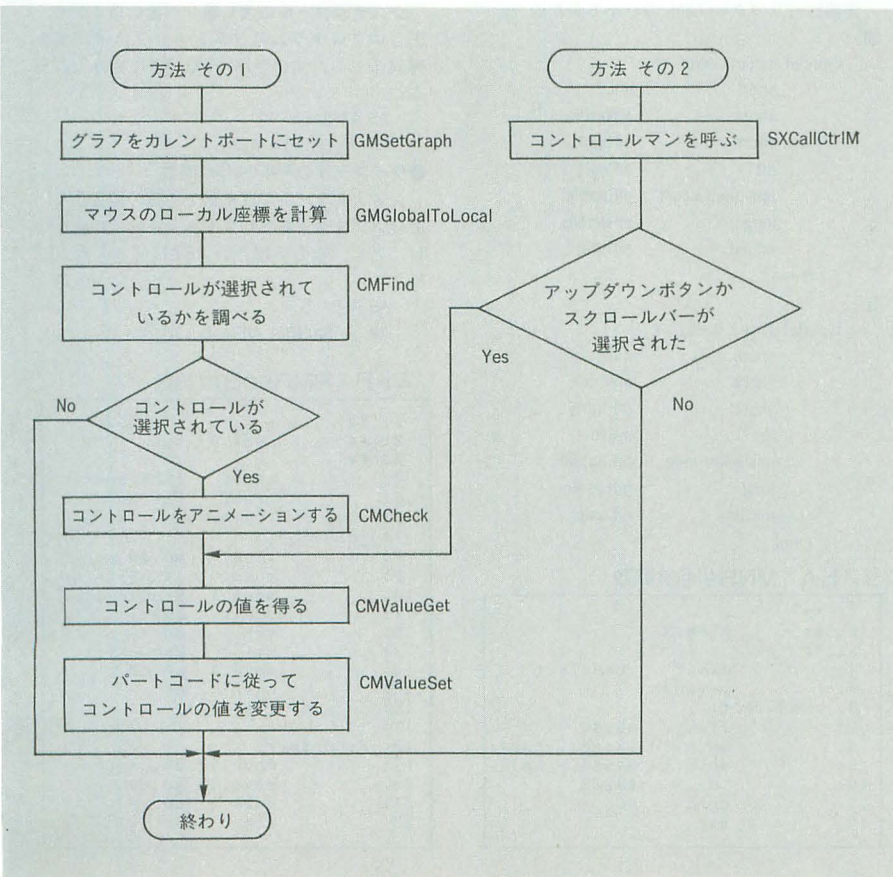
CtrlPrepare

と，

CtrlDispose

という関数をあらかじめ用意しています（アップデートイベントの処理ではもうすでにCMDraw関数を呼ぶようになっている）。

スケルトンプログラムではこれらの関数はただFALSEという値を返すだけですが，制御ボタンを使う場合はこれらの関数を書き換えるようにします。制御ボタンをオープンする処理はCtrlPrepareに記述します。これはオープンが成功したらTRUEを，失敗したらFALSEを返すような仕様にしています。また，CtrlDisposeにはオープンしているすべての制御ボタンをクローズする処理を記述します。

リスト3に制御ボタンを扱うサンプルプログラムのスケルトンプログラム（リスト2）に対する差分および変更する関数を示してあります（これだけをコンパイルしても実行できませんよ）。これは2つの制御ボタン（いまはどちらも標準ボタン）をウィンドウ上に置き，押されたボタンをウィン

ドウに書き込むプログラムです。

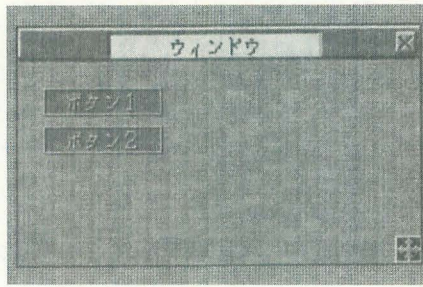
マウス左ボタンダウン時のイベント処理(procMSLDOWN関数)ではどちらのボタンが押されたかを書き込む処理が加わっていますが構造的には図2に示したとおりの操作をしているのがわかると思います(標準ボタンなので値の変更はしていません)。リスト3ではデフォルトで第1の方法を,

CMANAGER

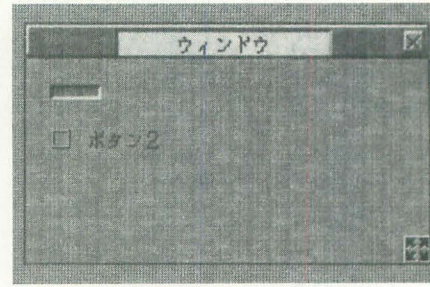
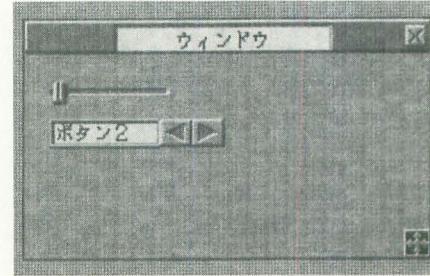
というシンボルが定義されていると第2の方法を使うように#ifdefが入っています。

標準ボタンのサンプルだけでは少しものたりないのでもう少し別のサンプルも紹介しましょう。基本的にはリスト3の,

```
#define CTRLID
#define CTRLMIN
#define CTRLMAX
      ⋮
```

という制御ボタンの定義の変更だけで間に合うのですが制御ボタンの値を変更するためにマウス左ボタンダウン時のイベント処理(procMSLDOWN関数)も少し変更する必要があります。リスト4とリスト5にリスト3に対する定義部とprocMSLDOWN関数の変更部分を示します。リスト4はセレクトボタンとオルタネートボタン,リスト5はスライドボリュームとアップダウンボタンを表示するサンプルです。

＊

ただ眺めてみるしかなかったSX-WINDOW上のアプリケーションも制御ボタンをつけることで人間からの指示を通知できるようになります。セレクトボタンやスライドボリュームなどを上手に使えばアプリケーションの自由度が上がることは間違いありません。皆さんも自分のプログラムに制御ボタンを付け加えてみましょう。さて,次回(あえて来月とは書かない)はスクロールバーに関して説明したいなと思っています(それまでに理解できれば……)。

サンプル1の実行結果

サンプル2の実行結果

サンプル3の実行結果

≪参考文献≫

1) 村田敏幸,「ウィンドウプログラミングへの道(2) C言語によるプログラミング」,Oh!X 1991年2月号,100-105pp.

2) 中森 章,「SXLIFE Part II ポップアップメニューの追加」,Oh!X 1991年2月号,116-119pp.

今月のバグ出し

プログラムを作る過程で,1月号で配布したヘッダファイルやC用のライブラリ内でいくつかのバグや不都合に気づいたのでここで修正しておきましょう。

●SXDEF.H内の16進表記

グラフマンのペンモード用の定数の16進定数が$で始まる表記になっている。以下のように変更する。

誤)

```
#define G_FORE    $000
#define G_BACK    $100
#define G_PPAT    $200
#define G_EPAT    $300
```

正)

```
#define G_FORE    0x000
#define G_BACK    0x100
#define G_PPAT    0x200
#define G_EPAT    0x300
```

●SXDEF.HのmlList型の定義

そのまま使用するとC言語でメニューを記述するのが困難になる。はたしてバグなのかは不明だが次のように書き換えた。

誤)

```
typedef  struct mlList  {
        short   mlLSize;
        mltem   mlLData;
}  mlList;
```

正)

```
typedef char    Str255 [255];
typedef struct  mlList  {
        short mlLSize;
        Str255 mlLData;
}  mlList;
```

●SXDEF.Hのmenu型の定義

使用許可フラグmEnableのバイト長が違う。

誤)

```
typedef struct menu  {
        short           mID;
        short           mWidth;
        short           mHight;
        int             mProc;
        unsigned short  mEnable;
        long            mHandle;
        mlList          mData;
}  menu;
```

正)

```
typedef struct menu  {
        short           mID;
        short           mWidth;
        short           mHight;
        int             mProc;
        unsigned long   mEnable;
        long            mHandle;
        mlList          mData;
}  menu;
```

リストA MNSelect関数

```
 1: **
 2: **        A268.S
 3: **
 4:           .xdef    _MNSelect
 5:           .even
 6: _MNSelect:
 7:           link     a6,#0
 8:           move.l   12(a6),-(sp)
 9:           move.l   8(a6),-(sp)
10:           .dc.w    $a268
11:           unlk     a6
12:           rts
```

●ライブラリのMNSelect関数

システムコールに渡す第2引数の値を誤る。正しいプログラムはリストA。ファイル名をA268.Oとし,次の手順でSXLIB.Aに組み入れること。

```
AS A268.S
AR /u SXLIB.A A268.O
```

●ライブラリのRMRscGet関数

システムコールに渡す第2引数の値を誤る。正しいプログラムはリストB。ファイル名をA0E1.Sとし,次の手順でSXLIB.Aに組み入れること。

```
AS A0E1.S
AR /u SXLIB.A A0E1.O
```

リストB RMRscGet関数

```
 1: **
 2: **        A0E1.S
 3: **
 4:           .xdef    _RMRscGet
 5:           .xref    _errno
 6:           .even
 7: _RMRscGet:
 8:           link     a6,#0
 9:           move.w   14(a6),-(sp)
10:           move.l   8(a6),-(sp)
11:           .dc.w    $a0e1
12:           tst.l    d0
13:           bmi      RscGetErr
14:           move.l   a0,d0
15:           unlk     a6
16:           rts
17:
18: RscGetErr:
19:           move.l   d0,_errno
20:           move.l   a0,d0
21:           unlk     a6
22:           rts
```

リスト2 スケルトンプログラム

```
 1: /*-
 2:
 3:     SX-WINDOWスケルトンプログラム
 4:
 5:                (C) 中森  章, Feb. 3, 1991
 6: */
 7: #include <stdio.h>
 8: #define __POINT_T        /* point_t 型を使う */
 9: #include <sxlib.h>
10: #define FALSE   0
11: #define TRUE    ~FALSE
12: /*
13:         ここでウィンドウに関する定数を設定
14: */
15: #define WDEFID          WI_STD
16: #define WINOPT          ( WC_GBOX | WC_GBOXON )
17: #define WINWIDTH        0x100
18: #define WINHIGHT        0x080
19: #define WINTITLE        "¥012ウィンドウ"
20: #define EVENTMASK       EM_EVERY
21:
22: #define MDEFID          1
23: #define MNENABLE        0xffffffff
24: #define MNITEMS         2
25: #define MNILIST         "¥0¥0¥013アイテム1 ¥0¥0¥013アイテム2 "
26: #define MNTITLE         "¥014メニューだよ"
27: /*
28:         ここは定数から計算される定数
29: */
30: #define WINOPTL         ( WINOPT & 0xf )
31: #define WINDEFID        ( WDEFID << 4 | WINOPTL )
32:
33: window  *winPtr;
34: rect    winSize;
35: event   eventRec;
36: int     activeFlag;
37:
38: int     ctrlFlag;
39: int     menuFlag;
40:
41: menu    **menuHdl;
42:
43: menu theMenu = {
44:         0,0,0,0,MNENABLE,0,{MNITEMS-1,MNILIST}
45: };
46:
47: main()
48: {
49:         if( SX_init()==FALSE ) OpenError() ;
50:         while( 1 ){
51:                 TSEventAvail(EVENTMASK,&eventRec);
52:                 switch( eventRec.eWhat ){
53:                 case E_IDLE:      procIDLE();     break;
54:                 case E_MSLDOWN:   procMSLDOWN(); break;
55:                 case E_MSLUP:     procMSLUP();   break;
56:                 case E_MSRDOWN:   procMSRDOWN(); break;
57:                 case E_MSRUP:     procMSRUP();   break;
58:                 case E_KEYDOWN:   procKEYDOWN(); break;
59:                 case E_KEYUP:     procKEYUP();   break;
60:                 case E_UPDATE:    procUPDATE();  break;
61:                 case E_ACTIVATE:  procACTIVATE();break;
62:                 case E_SYSTEM1:   procSYSTEM();  break;
63:                 case E_SYSTEM2:   procSYSTEM();  break;
64:                 case E_USER1:     procUSER();    break;
65:                 case E_USER2:     procUSER();    break;
66:                 }
67:         }
68: }
69:
70:
71: SX_init()
72: {
73:         task    taskBuf;
74:
75:         TSGetTdb(&taskBuf, -1);
76:         if( (TSTakeParam(&taskBuf.command,&winSize,NULL,0,NULL,NULL)&1)==0 ){
77:                 *(int *)&winSize.left = TSGetWindowPos();
78:                 winSize.right = winSize.left+WINWIDTH;
79:                 winSize.bottom= winSize.top +WINHIGHT;
80:         }
81:         winPtr=WMOpen(NULL,&winSize,WINTITLE,TRUE,WINDEFID,(window *)-1,TRUE,TSGetID());
82:         if( winPtr == NULL ) return( FALSE );
83:         winPtr->wOption = WINOPT;
84:         activeFlag=FALSE;
85:         ctrlFlag  = CtrlPrepare();/* コントロールが不要なら ctrlFlag=FALSE */
86:         menuFlag  = MenuPrepare();/* メニューが不要なら     menuFlag=FALSE */
87:         drawGrowBox();
88:         return( TRUE );
89: }
90:
91: SX_term()
92: {
93:         if( ctrlFlag ) CtrlDispose();
94:         if( menuFlag ) MenuDispose();
95:         WMDispose( winPtr );
96:         exit();
97: }
```

```
 98:
 99: drawGrowBox()
100: {
101:         GMSetGraph( winPtr );
102:         WMDrawGBox( winPtr );
103: }
104:
105: CtrlPrepare()
106: {
107:         return( FALSE );
108: }
109:
110: CtrlDispose()
111: {
112:         return( FALSE );
113: }
114:
115: MenuPrepare()
116: {
117:         menuHdl=(menu**)MMChHdlNew( sizeof(theMenu) );
118:         if( menuHdl == NULL ){
119:                 DMError(0x101,"メニュー領域が確保できません");
120:                 return ( FALSE );
121:         }
122:         memcpy(*menuHdl,&theMenu,sizeof(theMenu));
123:         (*menuHdl)->mProc=RMRscGet(('M'<<24)|('D'<<16)|('E'<<8)|'F',MDEFID);
124:         if( (int)((*menuHdl)->mProc)<=0 ){
125:                 MMHdlDispose(menuHdl);
126:                 DMError(0x101,"メニューがオープンできません");
127:                 return( FALSE );
128:         }
129: #if MDEFID==1
130:         (*menuHdl)->mHandle=MNTITLE;
131: #endif
132:         return( TRUE );
133: }
134:
135: MenuDispose()
136: {
137:         MMHdlDispose(menuHdl);
138:         return( TRUE );
139: }
140:
141: procIDLE()
142: {
143:         return( FALSE );
144: }
145:
146: procMSLDOWN()
147: {
148:         if( eventRec.eWhom != winPtr ) return( FALSE );
149:         if( activeFlag == FALSE ){
150:                 WMSelect( winPtr );
151:                 activeFlag = TRUE;
152:                 if( EMLStill() == 0){
153:                         TSGetEvent(EVENTMASK,&eventRec);
154:                         return( FALSE );
155:                 }
156:         }
157:         switch( SXCallWindM(winPtr,&eventRec) ){
158:         case W_INCLOSE:
159:                 SX_term(); break;
160:         case W_INGROW:
161:         case W_INZMOUT:
162:         case W_INZMIN:
163:                 GMClipRect(&winPtr->wGraph.grRect);
164:                 break;
165:         }
166:         TSGetEvent(EVENTMASK,&eventRec);
167:         return( TRUE );
168: }
169:
170: procMSLUP()
171: {
172:         return( FALSE );
173: }
174:
175: procMSRDOWN()
176: {
177:         int item;
178:         char BUF[128];
179:
180:         if( eventRec.eWhom != winPtr ) return( FALSE );
181:         GMSetGraph( winPtr );
182:         if( activeFlag == FALSE ){
183:                 WMSelect( winPtr );
184:                 activeFlag = TRUE;
185:                 if( EMRStill() == 0){
186:                         TSGetEvent(EVENTMASK,&eventRec);
187:                         return( FALSE );
188:                 }
189:         }
190:         item=MNSelect(menuHdl,eventRec.eWhere);
191:         TSGetEvent(EVENTMASK,&eventRec);
192:         sprintf(BUF,"メニュー番号は %d¥0",item);
193:         DMError(1,BUF);
194:         return( TRUE );
195: }
196:
197: procMSRUP()
```


▶「ライフゲームで姓名判断」いけますねえ。ちなみに私もやってみましたが,「藤」の字がネックらしく,どうもうまくいきません(なにが?)。複雑な漢字の名前の皆さん,要チャレンジですね。　佐藤　哲哉(27)東京都

```
198: {
199:         return( FALSE );
200: }
201:
202: procKEYDOWN()
203: {
204:         return( FALSE );
205: }
206:
207: procKEYUP()
208: {
209:         return( FALSE );
210: }
211:
212: procUPDATE()
213: {
214:         if( eventRec.eWhom != winPtr ) return( FALSE );
215:         WMUpdate( winPtr );
216:         if( ctrlFlag ) CMDraw( winPtr );
217:         WMUpdtOver( winPtr );
218:         drawGrowBox();
219:         TSGetEvent(EVENTMASK,&eventRec);
220: }
221:
222: procACTIVATE()
223: {
224:         if( eventRec.eWhom == winPtr )    activeFlag = TRUE;
225:         else if( eventRec.eWhom != NULL ){
226:                 if( activeFlag ) {
227:                         activeFlag = FALSE;
228:                         TSGetEvent(EVENTMASK,&eventRec);
229:                 }
230:         }
231:         return( TRUE );
232: }
233:
234: procSYSTEM()
235: {
236:         switch( ((tsevent*)&eventRec)->what2 ){
237:         case CLOSEALL:
238:         case ENDTSK:
239:                 SX_term(); break;
240:         case WINDOWSELECT:
241:                 WMSelect( winPtr ); break;
242:         }
243: }
244:
245: procUSER()
246: {
247:         return( FALSE );
248: }
249:
250: OpenError()
251: {
252:         DMError(0x101,"ウィンドウがオープンできません");
253:         SX_term();
254: }
```

リスト3　サンプル1

```
 1: /*
 2: ****************************************************************
 3: * スケルトンプログラムの先頭に追加
 4: ****************************************************************
 5: */
 6: #define CTRLID          CI_STDBTN             /* 標準ボタン */
 7: #define CTRLMIN         0                     /* 最小値     */
 8: #define CTRLMAX         1                     /* 最大値     */
 9: #define CTRLVAL         0                     /* 初期値     */
10: #define CTRLTITLE       "¥010ボタン1"         /* タイトル   */
11: #define CTRLLEFT        16                    /* 左上X座標 */
12: #define CTRLTOP         16                    /* 左上Y座標 */
13: #define CTRLRIGHT       CTRLLEFT+76           /* 右下X座標 */
14: #define CTRLBOTTOM      CTRLTOP +18           /* 右下Y座標 */
15:
16: #define CTRL2ID         CI_STDBTN             /* 標準ボタン */
17: #define CTRL2MIN        0                     /* 最小値     */
18: #define CTRL2MAX        1                     /* 最大値     */
19: #define CTRL2VAL        0                     /* 初期値     */
20: #define CTRL2TITLE      "¥010ボタン2"         /* タイトル   */
21: #define CTRL2LEFT       16                    /* 左上X座標 */
22: #define CTRL2TOP        40                    /* 左上Y座標 */
23: #define CTRL2RIGHT      CTRL2LEFT+76          /* 右下X座標 */
24: #define CTRL2BOTTOM     CTRL2TOP +18          /* 右下Y座標 */
25:
26: control **ctrlHdl;      /* コントロールへのハンドル      */
27: rect    ctrlSize;       /* コントロールの大きさ          */
28: int     ctrlValue;      /* コントロールの値              */
29:
30: control **ctrl2Hdl;     /* コントロールへのハンドル      */
31: rect    ctrl2Size;      /* コントロールの大きさ          */
32: int     ctrl2Value;     /* コントロールの値              */
33:
34: control **ctrlSelHdl;   /* 選択されたコントロールを格納 */
35:
36: /*
37: **************************************************************************
```

▶日電厚生サービス府中店（要はNEC内の売店）で，「X68000マシン語プログラミング」という本を見ました。「MS-DOS……」,「NOTE……」,「一太郎……」という本の中で「マシン語」の4文字は光っていました。私はこの売店にOh！Xを毎月予約しています（ちなみにOh！Xは意外と多い）。　佐川　正人(22)東京都

```
 38: * スケルトンプログラムから変更
 39: ***********************************************************************
 40: */
 41: CtrlPrepare()
 42: {
 43:         ctrlSize.left   =CTRLLEFT;
 44:         ctrlSize.top    =CTRLTOP;
 45:         ctrlSize.right  =CTRLRIGHT;
 46:         ctrlSize.bottom=CTRLBOTTOM;
 47:         ctrlHdl=CMOpen(winPtr,&ctrlSize,CTRLTITLE,TRUE,CTRLVAL,CTRLMIN,CTRLMAX,
 48:                         (CTRLID<<4), 0 );
 49:         if(ctrlHdl==NULL){
 50:                 DMError(0x101,"コントロールがオープンできません");
 51:                 return ( FALSE );
 52:         }
 53:
 54:         ctrl2Size.left   =CTRL2LEFT;
 55:         ctrl2Size.top    =CTRL2TOP;
 56:         ctrl2Size.right =CTRL2RIGHT;
 57:         ctrl2Size.bottom=CTRL2BOTTOM;
 58:         ctrl2Hdl=CMOpen(winPtr,&ctrl2Size,CTRL2TITLE,TRUE,CTRL2VAL,CTRL2MIN,CTRL2MAX,
 59:                         (CTRL2ID<<4), 0 );
 60:         if(ctrl2Hdl==NULL){
 61:                 CMDispose(ctrlHdl);
 62:                 DMError(0x101,"コントロールがオープンできません");
 63:                 return ( FALSE );
 64:         }
 65:         return( TRUE );
 66: }
 67:
 68: CtrlDispose()
 69: {
 70:         CMDispose(ctrlHdl);
 71:         CMDispose(ctrl2Hdl);
 72:         return( TRUE );
 73: }
 74:
 75: procMSLDOWN()
 76: {
 77:         int part;
 78:         point_t pt;
 79:         char  BUF[100];
 80:
 81:         if( eventRec.eWhom != winPtr ) return( FALSE );
 82:         if( activeFlag == FALSE ){
 83:                 WMSelect( winPtr );
 84:                 activeFlag = TRUE;
 85:                 if( EMLStill() == 0){
 86:                         TSGetEvent(EVENTMASK,&eventRec);
 87:                         return( FALSE );
 88:                 }
 89:         }
 90:         switch( SXCallWindM(winPtr,&eventRec) ){
 91:         case W_INCLOSE:
 92:                 SX_term(); break;
 93:         case W_INGROW:
 94:         case W_INZMOUT:
 95:         case W_INZMIN:
 96:                 GMClipRect(&winPtr->wGraph.grRect);
 97:                 break;
 98:         }
 99:
100: #ifdef CMANAGER
101:         part=SXCallCtrlM(winPtr,&eventRec,NULL,NULL,NULL,&ctrlSelHdl);
102: #else
103:         GMSetGraph(winPtr);
104:         pt.x_y=GMGlobalToLocal(eventRec.eWhere);
105:         part=CMFind(pt,winPtr,&ctrlSelHdl);
106:         if(ctrlSelHdl==ctrlHdl || ctrlSelHdl==ctrl2Hdl)
107:                                 part=CMCheck(ctrlSelHdl,pt,NULL);
108: #endif
109:         pt.p.x=16;
110:         pt.p.y=80;
111:         SXErase(pt.p.x , pt.p.y , 20);
112:         GMMove(pt);
113:         if(ctrlSelHdl==ctrlHdl){
114:                 ctrlValue=CMValueGet(ctrlHdl);
115:                 sprintf(BUF,"ボタン1が押された(値:%d)¥0",ctrlValue);
116:                 GMDrawStrZ(BUF);
117:         }
118:         else if(ctrlSelHdl==ctrl2Hdl){
119:                 ctrl2Value=CMValueGet(ctrl2Hdl);
120:                 sprintf(BUF,"ボタン2が押された(値:%d)¥0",ctrl2Value);
121:                 GMDrawStrZ(BUF);
122:         }
123:         TSGetEvent(EVENTMASK,&eventRec);
124:         return( TRUE );
125: }
126:
127: /*
128: ***********************************************************
129: * スケルトンプログラムに追加
130: ***********************************************************
131: */
132: SXErase(x,y,n)
133: int x;
134: int y;
135: int n;
136: {
137:         rect    prect;
```


▶研究室への配属が情報処理になった。68030のUNIXでプログラムを組むことができてラッキー。だが，別に大学でなくてもこんなことはできるんだよな。
奥田 健児(22)神奈川県

```
138:         int     pmode;
139:         pmode=GMPenMode(G_BACK|G_PSET);
140:         prect.left=x;
141:         prect.top=y;
142:         prect.right=x+12*n;
143:         prect.bottom=y+12;
144:         GMFillRect(&prect);
145:         GMPenMode(pmode);
146: }
147:
148: SXPrintN(x,y,n)          /* デバッグ用：数値のプリント */
149: int x;
150: int y;
151: int n;
152: {
153:         char BUF[50];
154:         point_t pt;
155:         pt.p.x=x;
156:         pt.p.y=y;
157:         sprintf(BUF,"%d¥0",n);
158:         GMMove(pt);
159:         GMDrawStrZ(BUF);
160: }
```

リスト4 サンプル2

```
 1: /*
 2: ****************************************************************
 3: * リスト５の差分・追加からの変更点
 4: ****************************************************************
 5: */
 6: #define CTRLID          CI_SELBTN               /* セレクトボタン */
 7: #define CTRLMIN         0                       /* 最小値         */
 8: #define CTRLMAX         1                       /* 最大値         */
 9: #define CTRLVAL         0                       /* 初期値         */
10: #define CTRLTITLE       "¥010ボタン１"          /* タイトル       */
11: #define CTRLLEFT        16                      /* 左上Ｘ座標     */
12: #define CTRLTOP         16                      /* 左上Ｙ座標     */
13: #define CTRLRIGHT       CTRLLEFT+76             /* 右下Ｘ座標     */
14: #define CTRLBOTTOM      CTRLTOP +18             /* 右下Ｙ座標     */
15:
16: #define CTRL2ID         CI_OTNBTN               /* オルタネイトボタン */
17: #define CTRL2MIN        0                       /* 最小値             */
18: #define CTRL2MAX        1                       /* 最大値             */
19: #define CTRL2VAL        0                       /* 初期値             */
20: #define CTRL2TITLE      "¥010ボタン２"          /* タイトル           */
21: #define CTRL2LEFT       16                      /* 左上Ｘ座標         */
22: #define CTRL2TOP        40                      /* 左上Ｙ座標         */
23: #define CTRL2RIGHT      CTRL2LEFT+76            /* 右下Ｘ座標         */
24: #define CTRL2BOTTOM     CTRL2TOP +18            /* 右下Ｙ座標         */
25:
26: procMSLDOWN()
27: {
28:         int part;
29:         point_t pt;
30:         char  BUF[100];
31:
32:         if( eventRec.eWhom != winPtr ) return( FALSE );
33:         if( activeFlag == FALSE ){
34:                 WMSelect( winPtr );
35:                 activeFlag = TRUE;
36:                 if( EMLStill() == 0){
37:                         TSGetEvent(EVENTMASK,&eventRec);
38:                         return( FALSE );
39:                 }
40:         }
41:         switch( SXCallWindM(winPtr,&eventRec) ){
42:         case W_INCLOSE:
43:                 SX_term(); break;
44:         case W_INGROW:
45:         case W_INZMOUT:
46:         case W_INZMIN:
47:                 GMClipRect(&winPtr->wGraph.grRect);
48:                 break;
49:         }
50: #ifdef CMANAGER
51:         part=SXCallCtrlM(winPtr,&eventRec,NULL,NULL,NULL,&ctrlSelHdl);
52: #else
53:         GMSetGraph(winPtr);
54:         pt.x_y=GMGlobalToLocal(eventRec.eWhere);
55:         part=CMFind(pt,winPtr,&ctrlSelHdl);
56:         if(ctrlSelHdl==ctrlHdl || ctrlSelHdl==ctrl2Hdl){
57:                 part=CMCheck(ctrlSelHdl,pt,NULL);
58:                 ctrlValue=CMValueGet(ctrlSelHdl);
59:                 CMValueSet(ctrlSelHdl,ctrlValue^1);/* 値を０／１反転 */
60:         }
61: #endif
62:         pt.p.x=16;
63:         pt.p.y=80;
64:         SXErase(pt.p.x , pt.p.y , 20);
65:         GMMove(pt);
66:         if(ctrlSelHdl==ctrlHdl){
67:                 ctrlValue=CMValueGet(ctrlHdl);
68:                 sprintf(BUF,"ボタン１が押された（値：%d）¥0",ctrlValue);
69:                 GMDrawStrZ(BUF);
70:         }
71:         else if(ctrlSelHdl==ctrl2Hdl){
```

▶やっと謎が解けた。「copy」と入力すると，coxpyとなる現象の謎が……。なるほど，キーボードの構造上の問題だったんですね。でも，X68000PROのキーボードには対応していないチューンアップだということなので，私は悲しい……。　久野　智彦(18)静岡県

```
72:                 ctrl2Value=CMValueGet(ctrl2Hdl);
73:                 sprintf(BUF,"ボタン２が押された（値：%d）¥0",ctrl2Value);
74:                 GMDrawStrZ(BUF);
75:         }
76:         TSGetEvent(EVENTMASK,&eventRec);
77:         return( TRUE );
78: }
```

リスト5 サンプル3

```
 1: /*
 2: ***************************************************************
 3: * リスト５の差分・追加からの変更点
 4: ***************************************************************
 5: */
 6: #define CTRLID          CI_SLDVOL          /* スライドボリューム */
 7: #define CTRLMIN         0                  /* 最小値           */
 8: #define CTRLMAX         100                /* 最大値           */
 9: #define CTRLVAL         0                  /* 初期値           */
10: #define CTRLTITLE       "¥010ボタン１"     /* タイトル         */
11: #define CTRLLEFT        16                 /* 左上Ｘ座標       */
12: #define CTRLTOP         16                 /* 左上Ｙ座標       */
13: #define CTRLRIGHT       CTRLLEFT+76        /* 右下Ｘ座標       */
14: #define CTRLBOTTOM      CTRLTOP +18        /* 右下Ｙ座標       */
15:
16: #define CTRL2ID         CI_SLDBTN           /* アップダウンボタン */
17: #define CTRL2MIN        0                   /* 最小値           */
18: #define CTRL2MAX        100                 /* 最大値           */
19: #define CTRL2VAL        0                   /* 初期値           */
20: #define CTRL2TITLE      "¥010ボタン２"      /* タイトル         */
21: #define CTRL2LEFT       16                  /* 左上Ｘ座標       */
22: #define CTRL2TOP        40                  /* 左上Ｙ座標       */
23: #define CTRL2RIGHT      CTRL2LEFT+76+36     /* 右下Ｘ座標       */
24: #define CTRL2BOTTOM     CTRL2TOP +18        /* 右下Ｙ座標       */
25:
26: procMSLDOWN()
27: {
28:         int part;
29:         point_t pt;
30:         char  BUF[100];
31:
32:         if( eventRec.eWhom != winPtr ) return( FALSE );
33:         if( activeFlag == FALSE ){
34:                 WMSelect( winPtr );
35:                 activeFlag = TRUE;
36:                 if( EMLStill() == 0){
37:                         TSGetEvent(EVENTMASK,&eventRec);
38:                         return( FALSE );
39:                 }
40:         }
41:         switch( SXCallWindM(winPtr,&eventRec) ){
42:         case W_INCLOSE:
43:                 SX_term(); break;
44:         case W_INGROW:
45:         case W_INZMOUT:
46:         case W_INZMIN:
47:                 GMClipRect(&winPtr->wGraph.grRect);
48:                 break;
49:         }
50: #ifdef CMANAGER
51:         part=SXCallCtrlM(winPtr,&eventRec,NULL,NULL,NULL,&ctrlSelHdl);
52: #else
53:         GMSetGraph(winPtr);
54:         pt.x_y=GMGlobalToLocal(eventRec.eWhere);
55:         part=CMFind(pt,winPtr,&ctrlSelHdl);
56:         if(ctrlSelHdl==ctrlHdl || ctrlSelHdl==ctrl2Hdl)
57:                                 part=CMCheck(ctrlSelHdl,pt,NULL);
58: #endif
59:         pt.p.x=16;
60:         pt.p.y=80;
61:         SXErase(pt.p.x , pt.p.y , 20);
62:         if(ctrlSelHdl==ctrlHdl){
63:                 ctrlValue=CMValueGet(ctrlHdl);
64:                 sprintf(BUF,"ボタン１が押された（値：%d）¥0",ctrlValue);
65:                 GMMove(pt);
66:                 GMDrawStrZ(BUF);
67:         }
68:         else if(ctrlSelHdl==ctrl2Hdl){
69:                 ctrl2Value=CMValueGet(ctrl2Hdl);
70:                 switch(part){
71:                 case C_INUP:
72:                         if(ctrl2Value+1<=CTRL2MAX) ctrl2Value++;
73:                         break;
74:                 case C_INDOWN:
75:                         if(ctrl2Value-1>=CTRL2MIN) ctrl2Value--;
76:                         break;
77:                 }
78:                 CMValueSet(ctrl2Hdl,ctrl2Value);
79:                 sprintf(BUF,"ボタン２が押された（値：%d）¥0",ctrl2Value);
80:                 GMMove(pt);
81:                 GMDrawStrZ(BUF);
82:         }
83:         TSGetEvent(EVENTMASK,&eventRec);
84:         return( TRUE );
85: }
```

シミュレーションプログラミング入門 第5回

温泉とコンピュータのファジィな関係

今回は筆者の華門真人さんはゆっくりと湯泉につかりにいってしまいました。そのかわりに投稿者の野間恒毅さんが大きなプログラムを手にして，「ファジィ」について講義してくれます。そして，華門さんが話をまとめてくれます。

Kamon Masato 華門 真人

春の息吹きが感じられる季節になってきた，といいたいところではあるが，現実にはまだ寒さが厳しい。

そんな寒さの中にも，晴れた日のぼんやりとした青空や温もりに満ちた日差し，そして，冬の間の鬱屈を晴らすかのように空に向かって存分に自己主張をしている小さな梅の花に，微かな，けれど確実にそこまで来ている春の足跡はうかがえる。

でも，そんな浮かれた気分も，一陣の冷たい突風でどこへやら消え去ってしまう。こんな季節には温泉がよく似合う。露天風呂につかって，まだ来ぬ春に思いを寄せる。甘い気分に冷や水をかけんとする北風も，温泉につかった体まで冷やすことはできない。むしろ，頭を冷やし冴え切った気分にしてくれる。

う～ん，日本人だねぇ。というわけで，前回の予告どおり，今回は北国の温泉からお送りする。北国とはいっても栃木県は塩原温泉郷からである。外は一面の銀世界，音もなく降りしきる雪。

雪というものは実に摩訶不思議な存在である。雪は何も語らず，ひっそりと降るだけだけれど，見ていると不思議と情緒的な気分になってくる。まるで雪が僕の中にあった何かを解き放ったかのように。

雪は地上にあるすべてのものを覆い隠してしまう。美しいものでも，醜いものでも。そこに差別などない。中東に雪が降れば，すぐに戦いをやめただろうなあ，なんて考えたくもなる。

つらつらと考えていたら，また温泉に入りたくなってしまった。

考えてみれば温泉というものも実に日本人的なものだ。たしかにヨーロッパにも温泉（スパ）なるものはあるのだけれど，むこうのはだいぶニュアンスが違う。水着を着た転地治療みたいな感覚だ。第一，むこうではどっぷりとお湯につかるなんてことはない，蒸気を浴びるだけだ。

とはいえ，日本式の温泉が本当に体にいいのかといわれると，そこはもはやファジィの世界である。体は温まるし，なにか元気になったような気分にもなる。もっとも温泉に入った挙げ句，疲れきって倒れそうになることもよくあるが（それは湯あたりしているだけかも）。いずれにせよあの色つきのお湯の何が体にいいのか，ということになるとはっきりしたことは何もいえない。

温泉学者（いるの？）にいわせると，硫黄イオンが，とか，弱アルカリ性で……，ということになるのだろうが，個人的にはむしろ気分的なものが多分にあるような気がする。僕にいわせれば，温泉はなんともファジィな（曖昧な）代物なのだ。

温泉と同じように，ファジィという言葉も最近大流行している。まさしく老若男女１億総ファジィ時代といわんばかりのフィーバーぶり。ところがこの「ファジィ」という言葉，流行にもそれなりのワケがあるらしい。

Japanese

このファジィという言葉，さすがにもう意味ぐらいはご存じだろう。そう「曖昧な」という意味だ。従来のコンピュータは，すべてを０か１かで判断する。アルシオーネは美しいクルマか。答え，false(０)。ではアルシオーネは速いクルマか。答え，false（０），といった具合だ。

この０か１かといった思考に柔軟性を持たせようとして生まれたのが，ファジィだ。ファジィでは０と１の間に0.001から0.5，果ては0.999まで多くの値を持つ。前の例でいえば，「アルシオーネは美しいクルマか。答え，まあ知る人ぞ知るデボネアAMGよりははるかにマシだし，個人の趣味もあるから，0.4ぐらいかな。では次，アルシオーネは速いクルマか。答え，少なくともトラクターよりは速い。それにドライバーにもよるから0.7ぐらい」，ということになる。

ところで，この「曖昧な」という言葉，ある方面で非常によく用いられる。

日本人はすぐ話をごまかすし，彼らのいわんとすることは曖昧で何をいいたいのかさっぱりわからん，なんてね。

そう，日本でのファジィの大流行と，日本の文化というのは決して無関係ではないのだ。

もともとファジィ理論というものはアメリカ，UCLAのザデー(Zadeh)教授という人によって考えだされた。1965年のことだ。０（偽）でも，１（真）でもない，もっと曖昧なものを取り扱おうということで提案されたのがファジィネス（曖昧さ）というわけだ。

しかし，ファジィは決していいスタートを切ったわけではないらしい。むしろ，さんざん批判されたという。おそらくその裏には西洋の思考というものが働いていたのだろう。

もともと西洋では，割り切った考え方が一般的だ。「このプランは成功するか」という質問があった場合，答えは２つしかない。イエスかノーかだ。「たぶん」とか，「努力してみます」なんて答えは西洋の文化にはなじまないのだ。

一事が万事こんな具合なのだから，「曖昧な論理」なんておよそファジィなものが受け入れられるはずもない。実際，ファジィというものが受け入れられ，広く世界を騒がすようになったのはここ５，６年のこと

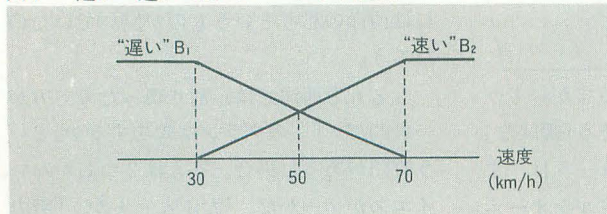
だ。

このファジィを広く普及させるのに大きな力を発揮したのがなにを隠そう「世界のいじめられっ子」日本なのである。言い換えれば日本的思考ということになるだろうか。「このプランは成功すると思うか」。多くの日本人は，まず成功しないだろうと思っていてもこう答えるに違いない。「う〜ん，たぶんなんとかなるでしょう」。

曖昧さをよしとする日本的思考は本質的にファジィネスの概念を内包しているのである。事実，ファジィ理論が実際に応用され，さまざまな分野で活用されているという面では，日本は疑う余地もなく世界一である。

最近では家電製品に飽き足らず，ファジィ国家日本などという呼び名まで登場する始末である。これがいいことなのかどうかは別としても，ファジィというものが日本の文化にいかにフィットしているのかはわかっていただけただろう。

こう考えてみると，日本人の温泉好きというものもわかってくるだろう(?)。効能が本当にあるのかどうかは問題でなく，単に温泉の雰囲気という目に見えない非常にファジィなものを楽しんでいるのだ。実のところ温泉とは日本人のファジィネスの産物だったのである。

さて，温泉と日本人のファジィな関係が明らかになったところで，そのファジィとは結局どのようなものなのか，いざシミュレーションという武器を用いて，ファジィの真髄に迫ってみることにしよう。といいたいところだが，僕は温泉につかりにいく。そのかわりといってはなんだが，今回は強力な助っ人を用意した。宇宙の遥か彼方「投稿」星から，ファジィによる車速自動制御シミュレーションをひっさげてやってきてくれた野間恒毅君である。

彼は大学4年生(もう卒業?)，マイコン歴はすでに約9年，PC-1500，MZ-2000，PC-8801mkIIFRときて，現在はX68000 EXPERTユーザーなんだそうな。

では少し不安だが彼にあとを託し，僕は温泉にでも入ってこよう。えっ，露天風呂が冬季閉鎖中なの。そんなバカな……。

a letter

拝啓　Oh!X編集部様

先日，Oh!X12月号「コンピュータシミュレーションの世界」を読ませていただきました。その中で連続変化モデルである車速制御のプログラムについて書いてありましたが，私も交差点のシミュレータを作ろうとして車速制御のプログラムを作っていました。そこでお送りします。見てやってください。

●「ファジー」と「ファジィ」の曖昧な関係

最近流行の言葉ですね。しかしてその実態はというと，その知名度とは裏腹に意外と知られていません。そこで問題です。「ファジー」と「ファジィ」の違いはなんでしょう。そう，それは曖昧ですが明確にわけられるのです。前者はマスコミおよびメーカーが作り上げたイメージであり，後者は技術者が処理を行うときに用いる手法なのです。さて，今回はこの「ファジィ」を使って車速制御プログラムを作ってみました。

早速，アルゴリズムの説明をしましょう。

●アルゴリズム

もともと，この「ファジィ」を考案したザデー教授は現代制御学を専門にしていらした方だそうで，その過程で「ファジィ」を考案しました。そのせいか，フィードバックのあるクローズされた制御系では非常にうまくいきます。これを利用して車間距離と速度をパラメータに与えるだけで，アクセル操作である加速度が返ってくる関数を作りました。今回発表するシステムはこの関数が作れるぐらいのファジィ操作関数を提供します。

さて，ひと口にファジィ制御とはいっても，実際にはどのようなことを行うのでしょうか。次のようなことが制御のルールとして挙げられます。

規則1：車の車間距離が狭くて車の速度が遅いなら，アクセルを保持せよ(速度を維持せよ)。

規則2：車の車間距離が狭くて車の速度が速いなら，ブレーキをかけよ（減速せよ)。

規則3：車の車間距離が広くて車の速度が遅いなら，アクセルを踏め（加速せよ)。

規則4：車の車間距離が広くて車の速度が速いなら，アクセルを保持せよ(速度を維持せよ)。

ところが，いままでの制御のルールですと，どの速度が速いか，または遅いかがわかりません。たとえば

　　if (速度 > 50 km/h) then 速い
　　　　　　　　　　　　　else 遅い

といったルールでは，50km/hを越えたとたんに急減速し，50km/hより遅くなったら，急加速するといったギクシャクしたものになるのは明らかです。車や家にあるエアコンの温度設定を思いだせばわかりやすいでしょう。設定温度の近くで冷やしたり暖めたりして，そのたびに思わず自分で設定温度を変えてしまいますよね。これじゃあなんのためのコンピュータ制御なのだろうかと疑問になります。

それではなぜそうなるのでしょうか。それは先ほどのif文でわかるように，速い・遅いの判別基準が一意だからです。その判別

図1　"速い""遅い"のメンバーシップ関数

図2　車間距離と加速度のメンバーシップ関数

基準を曖昧に，すなわち，なだらかにしたのがファジィのメンバーシップ関数です（図１）。

メンバーシップ関数は０から１までの値をとります。それは各々の速さ，遅さの度合いを示すわけです。たとえば，車速が40km/hですとメンバーシップ関数から速さの度合いは0.25，遅さの度合いは0.75と得られます。それでは次に車間距離と加速度のメンバーシップ関数を示します（図２）。

それではこれらのメンバーシップ関数をどのように先ほどの規則に当てはめるのでしょうか。規則１では「車の車間距離が狭くて車の速度が遅いなら……」とあります。この部分を規則の前件部といいます。それに対して「アクセルを保持せよ」の部分を後件部といいます。この前件部とどれだけ適合しているかの度合いが後件部の評価になります。

ここで例として，

車間距離：15m

速度　　：60km/h

のときを考えてみましょう。車間距離15mは先ほどの「車間が狭い」を表すメンバーシップ関数では0.75となります。速度60km/hは「速度が遅い」というメンバーシップ関数では0.25となります。つまり，車間距離15mは0.75程度狭く，速度60km/hは0.25程度遅いということを表します。さて，前件部では「……狭くて……なら」とありますが，それを言い換えると「……狭くてなおかつ……なら」となります。

それを踏まえて規則１を一般的な条件文にさらに直すと

if (“車間が狭い” and “速度が遅い”)
then 速度を保持せよ

といったand演算となります。いわゆる一般のand演算は両方が成り立つとき正となりますが，メンバーシップ関数の場合は０から１までの値になりますので，メンバーシップ関数のand演算は通常と異なり一般的にどちらか一方の小さい値をとります。これをmin演算といいます。この処理の方法は人によって違い，簡単に小さいほうをとらずに計算するやり方もあるようですが，ここでは一般的なmin演算を用いました。これをいまの例に当てはめると，車間の狭さが0.75，速度の遅さが0.25ですから，小さいほう，すなわち，速度の遅さの値をとり，前件部の適合度は0.25になります。この適合度分だけ後件部が成り立つと考えてください。

この処理を規則４まで行うと次のような適合度が得られます。

規則１：速度を保持せよ 0.25
規則２：減速せよ 0.75
規則３：加速せよ 0.25
規則４：速度を保持せよ 0.25

ここで考えなければならないのは適合度です。「速度を保持せよ」という命令のメンバーシップ関数は０から１までの値を持っていますが，適合度が0.25の場合は0.25から上の部分は意味を持ちません。なぜなら，たった0.25しかあっていないものをそのまま命令されては困るからです。控えめに命令してもらわなければなりません。ですから，ここでの適合度はいわば「控えめ度」もしくは「実行度」と考えるとわかりやすいでしょう。

そこで，控えめにするために先ほどのメンバーシップ関数に処理を行う必要が生じます。適合度0.25の場合，0.25より大きい部分は意味を持たないようにするために，0.25より大きい部分はすべて0.25にしてしまうのです。つまり本当は0.25以上の意味合いをもつのだが，適合度が0.25なので0.25以上は確かでない。この考えをメンバーシップ関数にすると図３のようになります。この処理をαカットといいます。ここではα＝0.25となっています。

このαカットを行った後件部のメンバーシップ関数が得られました(図４)。この４つの後件部をひとつにまとめなければ最終的なアクセル操作は行えません。これをまとめる前に，もう１回規則１から４を振り返ってみましょう。規則はどれかひとつだけが成り立てばそれを実行するのではないですね。もし，ひとつしか成り立たないのであれば，加速なら（程度は変わっても）加速しっぱなしだし，減速なら減速しっぱなしです。逆にいえばひとつだけが，ではなくこの４つの規則のバランスのうえで最終的な規則がまとめられるわけです。そうすると規則を結びつけているのは，orであることがなんとなくわかっていただけるでしょう。

そこで今度は４つの規則をまとめるためのメンバーシップ関数のor演算ですが，これは先ほどのand演算がminをとるのに対し，maxをとります。つまり，大きなほうをとるというわけです（これについてもmin演算と同様にいろいろな演算方法が提唱されてます）。それではmax演算を後件部のメンバーシップ関数に対して行いましょう。先ほどのmin演算は単に２つの数字を（１次元的に）比べただけですが，今回は横方向に加速度(および減速度)，縦方向にファジィ数０から１までの２次元に対して処理します。基本的に大きい数字をとればいいわけですから，単純にこれらのメンバーシップ関数のグラフを重ねて大きな値の線をとれば出来上がりです。もちろん，min演算をやりたいのであれば，逆に小さい値の線をとればいいのです(図５)。

４つの後件部のメンバーシップ関数にmax演算を行うと最終的な後件部の結果が得られます(図６)。でも，まだ終わりではありません。演算結果はメンバーシップ関数で得られるので，それを値に直さなければ命令を与えることができませんね。それを決めるのに「重心法」を用います。名前のとおり，このメンバーシップ関数の全体の面積を計算し，重心を求めます。求まった重心の横方向の値が最終的な命令の操作量となるわけです。ここではアクセルの操作量なわけですね。

この例の操作量はだいたい－8.3333にな

図３　αカット

ります。つまり，かなりブレーキを踏んで，減速するということです。感覚的にあってると思いませんか？　もちろん，60km/hで15mの車間距離なんて長いと思っている人は別(どうやら僕は別らしい……。華門)ですが，シミュレーションとしては正しいと思います。

いままでの説明でおわかりいただけたでしょうか？　かなりはしょって説明したのでわかりづらいと思いますが，そういうときは文末の参考文献を見てください。個人的には向殿さんの本がわかりやすく，興味をひかれました。

●関数の説明

プログラムはすべてXCver.2.0で作りました。Cの関数ですべて処理をしています。よって，他機種への移植は一部のグラフィック関係を修正するだけで簡単に実現できます。私の場合，グラフィック命令にすべてパッチをあててPC-9801用に書き換えて使ってます。

ファジィ演算用の関数については表1にまとめてみました。参考にしてください。

●車速制御プログラムの説明

メイン関数の最初の部分でメンバーシップ関数の定義を行ってます。「規則登録」ではアルゴリズムの説明であったように4つの規則を登録してます。そしてループ中で初期値を入力し，関数simuを呼びます。関数simuでは実際の制御を行います。実際のシミュレーションはたった1行，

a=fuz_est (l, v);

で行っています。この関数だけで先ほどのアルゴリズムで行ったことすべてをやっているわけです。次の行からはシミュレーションである計算部分です。単位系は速度を中心としたため距離がkm，時間がhとなってます。すなわち，速度はkm/h，加速度はkm/h²となってます。加速度の調整（1000倍）を行っているのは，参考文献でのメンバーシップ関数の値の最大値が±12.5km/h²となっていたためです。作ってる最中は気づかなかったのですが，1時間加速してたったの12.5km/hしか速度が上がらないなんてのはおかしいですよね。

そこで常識的な車の加速度を計算したところ，だいたい1000倍すればいいのではなかろうかということでこうなりました（当然，メンバーシップ関数の横軸である加速度を1000倍しても同じ結果が得られる）。

実際にシミュレートしてみればわかるように，いい線いってると思います。ただし遅すぎることと，とんでもない入力値を入れると衝突しますので注意してください。時間の単位は小さいほうが細かい制御をしてくれるので，衝突しづらくなります。また，スカイラインGT-Rなどの加速にするにはこの値を3000にしてください。とんでもない加速をして，すぐさま減速するはずです。もっとも本来ならGT-Rはほかの車の後ろを走る車じゃないですから（悲しいかな，よく見かける渋滞列の中のGT-R。華門)。

●特徴と問題点

ファジィコンピュータシミュレータとしてOh!X1990年4月号の後藤裕治さんの「I-MY」が発表されていますが，このプログラムは離散型のメンバーシップ関数を使用してます。一方，今回のものは連続型のメンバーシップ関数を使用しています。

この2つの違いは，離散型はメンバーシップ関数がバラバラの数値で与えられるため，速いが比較的大雑把で，連続型は逆に連続した関数として与えられるため，演算に時間がかかるが細かいということです。システムとしては「I-MY」は対話的にメンバーシップ関数を作ったり，演算したりしているようです。今回のものは対話的でなくCの関数としています。これは応用を目的としていますのでこうしました。プログラムの中に埋め込んで使うことが前提なためです。

今回，向こうをはって(?)作ってみましたが，案の定速いとはいえませんね。なおかつ，プログラムが複雑で長い。作った本人も困ってます。いちばん苦労したのは関数fuz_orとfuz_andです。これに関してはデバッグにかなり時間がかかりました。途中でアルゴリズムまで変えました。よってまだバグがあるかもしれないので，見つかったら知らせてください。

そこまで苦労して連続型にこだわったのは，車速制御のプログラムのような細かい制御系に向いているからでしょう。なんといっても処理系をすべてdouble型で宣言しているだけはあります。もっともX68000のC言語の関数はすべてdouble型64ビットになってますので，わざわざfloat型にキャストしてもしょうがありません。というわけで，制御結果は必要以上にきめ細かくなっています。

一方，問題点としては遅いということでしょう。姑息にハードウェアの速度に頼って，同じプログラムをPC-9801RA+387(20MHz)で動かしたら10倍くらい（あくまでも気分であり実測ではない）速くなりました。ハードに頼らずに速くするには処理系をすべてint型，場合によってはshort int, char型を使用するといいのでしょうが。し

図4　各規則によって得られた制御メンバーシップ関数（ここでは加速度）

かし，わかっていても面倒臭くてやる気がおきない。ああ。

●交差点シミュレータへの発展

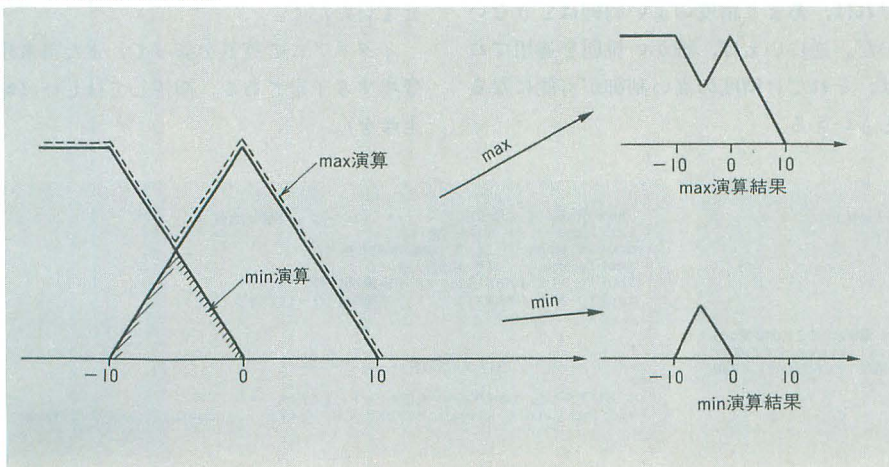
今回は車1台についての制御でしたが，これを同時に何台もやれば渋滞シミュレータや，交差点シミュレータに発展させることができるでしょう。しかし，このままX68000上で行うには苦しすぎます。原因としては，いまはメンバーシップ関数をdouble型として連続的に処理しており，いちいち規則に従って演算を行っていることにありそうです。

高速化するにはint型への変更，離散型への変更，および最終的には演算せずにn次関数（またはテーブル）にしてしまうのがてっとり早いと思います。結局，得られる値はいつでも不変ですから，テーブル参照型にしてしまえばかなりの速度向上が望めそうです（ただし汎用性はまったくなくなる）。誰かやってください。僕は卒業研究で忙しいのでいまはできそうにありません（もう終わっているんじゃ……。華門）。

●最後に

この関数群を使って，もうちょっとましな頭を持った「敵」を作ることもできるのではないでしょうか。いま自分で考えているのはブラックジャックの思考部分です。たとえばカードの合計が15だとするともう1枚ほしくなるが，親のカードが12であれば次に絵札を引いてドボンするかもしれない。こういったルールを自分なりに考えて合成するとなにかできると思います。

参考文献

・日刊工業新聞社「ファジィのはなし」，向殿政男著

・ラッセル社　ラッセルブックス「C言語による実用ファジィブックス」，三矢直城・田中一男 共著，p.77 - p.80

・日刊工業新聞社「ファジィ制御」，菅野道夫著，p.76 - p.82

それでは乱文乱筆乱プログラム，失礼しました。

草々

野間　恒毅

then

ああ，いいお湯だった。どうやらちょうど彼の講義が終わったようだね。どうだろう，皆さんファジィというものが見えてきただろうか。

まあ，たくさんの言葉よりも，プログラムを入力し，自分でいろいろと試してみるのがいちばんの近道だろうと思う。今回，編集部のご好意もあって，かなり大きなプログラムをすべて掲載することができた。

投稿原稿ではプログラムの内容まではあまり詳しく触れていないが，親切な彼のこと，コメント行を読めばだいたいは理解できるだろう。

長いプログラムだが，じっくり腰を落ち着けて入力してやれば，あとあと役に立つと思う。今回の車速制御プログラムは，このファジィエンジンのひとつの例でしかない。これをお手本にして，自分なりに，「ファジィ」なプログラムを書いてみるといいだろう。

おっと，肝心なプログラムはといえばメインプログラム（今回の車速制御プログラム）がリスト1「SPDX.C」であり，これをサポートするのがリスト2，ファジィ演算関数群「FUZ_MEM.C」，リスト3，ファジィ推論エンジン「FUZ_ENG.C」，リスト4，ファジィ規則登録部「FUZ_RULE.C」，リスト5，ファジィMAXMIN演算部「FUZ_OPR.C」，そしてヘッダファイルがリスト6「FUZ.H」ということになる。

さらにこれらをまとめてコンパイルするためのメイクファイルがリスト7「MK」である。ということは今回のプログラムを実行するためには約1700行も入力しなければならないことになる。

ま，骨の髄まで使ってやればそれだけの価値はあるはず，噛めば噛むほどってやつですな。

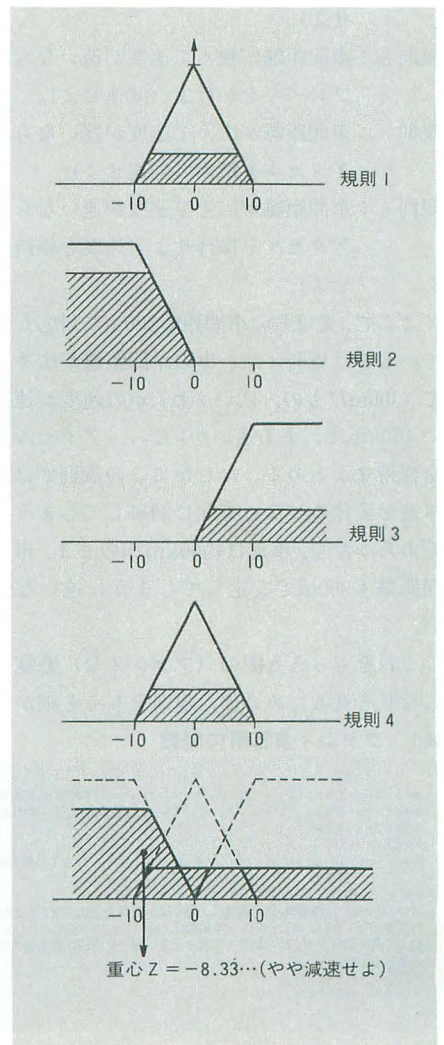
これらすべてを入力したら，C Compiler ver.2.00上で，

```
make -f mk
```

とすれば目的のSPDX.Xが得られるはずだ。

試しに実行してみよう，なになに，車間

図5　max演算とmin演算

図6　最終結果

距離，まあ20mぐらいかな，先行車はと，まあ60km/hぐらいか，それで後続車が，50km/hとしてみるか。

おお，たしかにファジィっぽい加速をするなあ。それじゃあ，後続車の車速を0にすると，どうだろう。あれ，あっという間においていかれてしまった。車速は同じ60km/hなんだけれど，車間距離がなんと300m。これはいくらなんでも広すぎ。

実はここにファジィの罠がある。最近はファジィばやりで，なんでもファジィの名前をつければ売れるようだ。でも，いくらファジィ制御とはいえども，ちゃんとした規則を持たなければあまりいい制御はできないのだ。

今回，車間距離が300mで安定してしまったのは，ファジィ制御の規則が不十分であったことによる。もう一度規則を思い出してみよう。

規則１：車間距離が狭くて速度が遅いなら，アクセルを保持せよ（速度を維持せよ）。

規則２：車間距離が狭くて速度が速いなら，ブレーキをかけよ（減速せよ）。

規則３：車間距離が広くて速度が遅いなら，アクセルを踏め（加速せよ）。

規則４：車間距離が広くて速度が速いなら，アクセルを保持せよ（速度を維持せよ）。

ここで，先ほどの車間距離300mに対応しているのは規則４だ。車の車間距離が広くて（300mだもの，広いよね）車の速度が速い（60km/h，まあ速いか）なら，アクセルを保持せよとある。たしかにこの規則では，車速を変化させない方向に制御してしまうであろうから，車速は約60km/hのまま，車間距離も300mで安定してしまうに違いない。

これをもっと人間の（ファジィな）感覚になじませるためには，規則をもっと細かくしてやる必要が出てくる。たとえば適正な車間距離をファジィ値で得るようにして，この適正値と現実の車間距離の差を計算し（当然これもファジィ値になる），その差（理想と現実の差……）をもとに加速，減速するようにすればよいことになる。

もっとも，こうすると規則がだいぶ複雑になってきてしまうのはいたしかたないところではある。しかし，考え方を変えれば，複雑なのはそれだけ細かい制御をしている証ともいえる（もちろん，不必要に複雑なのはムダそのものだが）。

仙台の地下鉄が車速のファジィ制御を行って話題になったが，この制御には安全性，乗り心地，電力消費，走行速度など６つのファジィ変数を用いて，実に24にものぼる規則を作り上げている。

実際，それだけの細かい制御を行っただけあって，かなり精度の高い制御を実現しているらしい。ところでこの24にものぼる規則がどこから出てきているかというと，これがなんと熟練運転手の経験からなんだそうである。

ま，それだけ人間の思考に近い表現ができる，という証である。

and……

結局のところ，ファジィとはそんなに革新的で，すごいものなんだろうか。たしかに，人間の思考パターンに近い表記ができるし，したがって，人間的な制御が可能になるということがおわかりいただけただろう。

しかし，だからといって「ファジィ＝賢い」とするのも考え物だろう。いままで見たように，制御の規則がしっかりしていなければ，あまり精度のよい制御はできないのだ。逆にいえば，細かい規則を適用すれば，それだけ精度の高い制御が可能になるともいえる。

前回，ファジィ制御はPID制御を超えると書いたが，そういった見方からすれば，これは正しいとも正しくないともいえることになりそうだ。さらにこの考え方を推し進めると，最近のファジィ家電もどうかなということになってくる。あれは本当にちゃんとした制御を行っているのだろうか。名前だけのファジィとかも多そうだけれど大いなる疑問である。

とはいえ，綿密な規則に基づいて行われるファジィ制御には通常の制御には期待できない高度な制御が期待できる。より人間に近い制御。使いこなしてやりさえすれば可能性はPID制御の比ではないといえそうだ。それじゃあ僕ももう一度温泉にでもつかって，ファジィの未来にでも思いをはせてみることにしよう。

NEXT

今回，ファジィに真っ向から取り組んでいたら，いい加減ファジィさに疲れてきてしまった（温泉などというファジィなものにつかりすぎたため，風邪をひいただけ，という話がある）。日本はファジィすぎる。コマーシャルにしても「友達以上恋人未満」などと実にファジィなもので満ち溢れている（古い話でもうしわけないが，「なんとなくクリスタル」という題名もいまから考えてみれば非常にファジィである）。

ヨーロッパの「イエス or ノー」というのも嫌だけれど，日本のようにあまりにファジィなのも疲れるものがある。

その中間ならばいいのに，ということでファジィなヨーロッパ，イタリアに逃亡を図ることにした。当然，来月はお休みにさせていただく。

イタリアにて英気を養って，また再来月登場する予定である。期待してほしい（お土産を）。

表1　ファジィ演算用の関数

```
#define CHARBUF  32  /* メンバーシップ関数の名前用キャラクタの最大バッファ */
#define DMAX  30  /* メンバーシップ関数の点データの最大数 */
#define ERR   0
#define OK    1
#define RULE_MAX 10  /* ファジィルールベースの最大数 */

/*
  メンバーシップ関数を表す構造体。最小値から最大値迄の横方向の値を持つ。原則としてこの制限値から
外れないで下さい。メンバーシップ関数はx[0],y[0]を始点として，x[num],y[num]までの点を
結ぶ直線で表現する。始点が最小値より大きい場合や終点が最大値より小さい場合，その間は0として処理
している。
*/
struct MEMBER_TAG {
```

```
  char name [ CHARBUF ]; /* メンバーシップ関数の名前 */
  int num;    /* データの数 */
  double min;    /* X（横軸）の最小値 */
  double max;    /* Xの最大値 */
  double x [ DMAX ];   /* X座標の値 */
  double y [ DMAX ];   /* Y座標の値（0～1のファジィ数）*/
};

/*
--------------プロトタイプ宣言--------------
*/
/* NOTをとる。1からYを引いた値をセットしてるだけ。 */
void fuz_not ( struct MEMBER_TAG *temp , struct MEMBER_TAG *mem
```

```
);
/* メンバーシップ関数の一番値の大きい場所(*address)をとり,値を返す。 */
double fuz_sup ( struct MEMBER_TAG mem , double *address );
/* 同様に一番小さいところをとる。 */
double fuz_inf ( struct MEMBER_TAG mem , double *address );
/* Xで指す所のY(ファジィ数)を返す。 */
double fuz_val ( double x , struct MEMBER_TAG mem );
/* mema , memb のORをとったものをtempに入れる。*/
void fuz_or ( struct MEMBER_TAG *temp , struct MEMBER_TAG *mema
, struct MEMBER_TAG *memb);
/* mema , memb のANDをとったものをtempに入れる。*/
void fuz_and ( struct MEMBER_TAG *temp , struct MEMBER_TAG *mema
 , struct MEMBER_TAG *memb);
/* メンバーシップ関数を表示する */
void fuz_graph ( int xs , int ys , struct MEMBER_TAG mem );
/* メンバーシップ関数をセーブする */
int fuz_save ( char *fname , struct MEMBER_TAG *mem );
/* メンバーシップ関数をロードする */
int fuz_load ( char *fname , struct MEMBER_TAG *mem );
/* メンバーシップ関数をコピーする */
void fuz_cpy ( struct MEMBER_TAG *temp , struct MEMBER_TAG *mem
);
/* 三角型のメンバーシップ関数を作る。centerを中心とし,2*rangeを底辺とする最大valの二
等辺三角形。Xの最小値,最大値はmin , max で与える。*/
void fuz_tri ( struct MEMBER_TAG *temp , double center , double
range
   , double val , double min , double max );
/* fuz_gravで使用。面積を求める。 */
double fuz_area ( double x1 , double y1 , double x2 , double y2
);
/* 重心を求める。 */
double fuz_grav ( struct MEMBER_TAG mem);
/* valでαカットする。*/
void fuz_acut ( struct MEMBER_TAG *temp , struct MEMBER_TAG *mem
 , double val );
/* メンバーシップ関数を縦方向にk倍する。*/
void fuz_scl ( double k , struct MEMBER_TAG *mem );
/* メンバーシップ関数を正規化(Yの最大値が1になるように定数倍)する */
void fuz_norm ( struct MEMBER_TAG *mem );
/* メンバーシップ関数をk乗する。 */
void fuz_beki ( double k , struct MEMBER_TAG *mem );
/* メンバーシップ関数の集中化(2乗) */
void fuz_con .( struct MEMBER_TAG *mem );
/* メンバーシップ関数の拡大化(1/2乗) */
void fuz_dil ( struct MEMBER_TAG *mem );
/* ファジィルールベースの登録。ルールベースは前件部2つ(memx,memy),後件部1つ(memz)
で構成している。返り値は登録件数 */
int fuz_rule ( struct MEMBER_TAG *memx , struct MEMBER_TAG *memy
 , struct MEMBER_TAG *memz );
/* 小さい方の値を返す */
double fuz_min ( double a , double b );
/* 大きい方の値を返す */
double fuz_max ( double a , double b );
/* ファジィ推論を行う。a , bは前件部へのデータ */
double fuz_est ( double a , double b );
/* ファジィ推論を途中結果を見せながら行う。a , bは前件部へのデータ。
 x,yはグラフ表示する始点。 */
double fuz_est_g ( double a , double b , int x , int y );
```

リスト1 SPDX.C

```
  1: /*
  2: ************************************************************************
  3:    SPDX     リアルタイム車速制御シミュレーション
  4:
  5:                      Programmed by T.Noma
  6:            ver 1.0          '90.10.21
  7:            ver 2.0          リアルタイム車速制御を行う
  8:            ver 2.1          グラフィックスを 綺 麗にする
  9:            ver 2.2          Oh!X用修正
 10:            ver 2.3          fuz_or,fuz_and関数デバッグ、fuz_opr.o付加
 11:    参考文献
 12:            ・ラッセル社 ラッセルブックス「C言語による実用ファジィブックス」
 13:                      三矢直城・田中一男 共著 p.77 - p.80
 14:            ・日刊工業新聞社 「ファジィ制御」
 15:                      菅野道夫 著  p.76 - p.82
 16:    Include File
 17:                      fuz.h
 18:    Link Files
 19:                      fuz_mem.o
 20:                      fuz_eng.o
 21:                      fuz_rule.o
 22:                      fuz_opr.o
 23: ************************************************************************
 24: */
 25: #include <stdio.h>
 26: #include <string.h>
 27: #include <graph.h>
 28: #include <basic0.h>
 29: #include "fuz.h"
 30:
 31: /*
 32: -------------グローバル変数-------------
 33:    だいたいファジィ・ルールベースを
 34:    意味する。最大値はRULE_MAX
 35: -----------------------------------------
 36: */
 37: int         rule_num ;
 38: struct      MEMBER_TAG rule_x [ RULE_MAX ] , rule_y [ RULE_MAX ] , rule_z [
RULE_MAX ];
 39:
 40: /*
 41: ****************************************************
 42: */
 43:
 44: void        simu ( l , v , vs , t )
 45: double      l , v , vs , t ;
 46: {
 47:    double  a;
 48:    char    s [ 8 ];
 49:    int     li , ll;           /* 描画用の距離、前の距離 */
 50:
 51:    ll = ( int ) l;
 52:    printf ( "a:%lf v:%lf l:%lf ¥n" , a , v , l );
 53:
 54:    fill ( 500 , 80 , 510 , 100 , 15 );      /*      先頭車の描画     */
 55:    while ( -1 ){
 56:            if ( l < 0 ){
 57:                    printf ( "¥n衝突しました。¥n" );
 58:                    break;
 59:            }
 60:            /*      ファジィ推論を行っている。たった一行だから簡単至極      */
 61:            a = fuz_est ( l , v );
 62:
 63:            v +=  a * 1000 * t;             /* 加速度の調整 */
 64:            l += ( vs - v ) * t * 1000;     /* km -> m に変換 */
 65:            li = ( int )( l * 4 );
 66:            printf ( "a:%lf l:%22.17lf v:%22.17lf ¥r" , a , l , v );
 67:            fill ( 500 , 100 + ll , 510 , 120 + ll , 0 );
 68:            fill ( 500 , 100 + li , 510 , 120 + li , 3 );
 69:
 70:            ll = li;
 71:            b_inkey0 ( s );
 72:            if ( *s == ' ' ){
 73:                    break;
 74:            }
 75:    }
 76: }
 77:
 78:
 79: /********** Main **************/
 80: void        main ()
 81: {
 82:    static  struct  MEMBER_TAG mema [ 2 ] , memb [ 2 ], memc [ 3 ] , memd ,
meme ;
 83:    double  sup , xs , inf , xi , n;
 84:    int     num;
 85:    double  l , v , vs , t ;
 86:
 87:    /*      規則:車間距離  */
 88:    strcpy ( mema [ 0 ].name , "規則:車間距離狭い" );
 89:    mema [ 0 ].num = 3;
 90:    mema [ 0 ].min = 0.0;
 91:    mema [ 0 ].max = 40.0;
 92:
 93:    mema [ 0 ].x [ 0 ] = 0.0;
 94:    mema [ 0 ].y [ 0 ] = 1.0;
 95:    mema [ 0 ].x [ 1 ] = 10.0;
 96:    mema [ 0 ].y [ 1 ] = 1.0;
 97:    mema [ 0 ].x [ 2 ] = 30.0;
 98:    mema [ 0 ].y [ 2 ] = 0.0;
 99:
100:    fuz_not ( &mema [ 1 ] , &mema [ 0 ] );
101:    strcpy ( mema [ 1 ].name , "規則:車間距離広い" );
102:
103:    /*      規則:車速       */
104:    strcpy ( memb [ 0 ].name , "規則:車速遅い" );
105:    memb [ 0 ].num = 3;
106:    memb [ 0 ].min = 0.0;
107:    memb [ 0 ].max = 100.0;
108:
109:    memb [ 0 ].x [ 0 ] = 0.0;
110:    memb [ 0 ].y [ 0 ] = 1.0;
111:    memb [ 0 ].x [ 1 ] = 30.0;
112:    memb [ 0 ].y [ 1 ] = 1.0;
113:    memb [ 0 ].x [ 2 ] = 70.0;
114:    memb [ 0 ].y [ 2 ] = 0.0;
115:
116:    fuz_not ( &memb [ 1 ] , &memb [ 0 ] );
117:    strcpy ( memb [ 1 ].name , "規則:車速速い" );
118:
119:    /*      制御:加速度     */
120:    strcpy ( memc [ 0 ].name , "制御:加速度そのまま" );
121:    memc [ 0 ].num = 3;
122:    memc [ 0 ].min = -20.0;
123:    memc [ 0 ].max = +20.0;
124:
125:    memc [ 0 ].x [ 0 ] = -10.0;
126:    memc [ 0 ].y [ 0 ] = 0.0;
127:    memc [ 0 ].x [ 1 ] = 0.0;
128:    memc [ 0 ].y [ 1 ] = 1.0;
129:    memc [ 0 ].x [ 2 ] = 10.0;
130:    memc [ 0 ].y [ 2 ] = 0.0;
131:
132:    strcpy ( memc [ 1 ].name , "制御:加速度減速" );
133:    memc [ 1 ].num = 3;
134:    memc [ 1 ].min = -20.0;
135:    memc [ 1 ].max = +20.0;
```

```
136:
137:    memc [ 1 ].x [ 0 ] = -20.0;
138:    memc [ 1 ].y [ 0 ] = 1.0;
139:    memc [ 1 ].x [ 1 ] = -10.0;
140:    memc [ 1 ].y [ 1 ] = 1.0;
141:    memc [ 1 ].x [ 2 ] = 0.0;
142:    memc [ 1 ].y [ 2 ] = 0.0;
143:
144:    strcpy ( memc [ 2 ].name , "制御：加速度加速" );
145:    memc [ 2 ].num = 3;
146:    memc [ 2 ].min = -20.0;
147:    memc [ 2 ].max = +20.0;
148:
149:    memc [ 2 ].x [ 0 ] = 0.0;
150:    memc [ 2 ].y [ 0 ] = 0.0;
151:    memc [ 2 ].x [ 1 ] = 10.0;
152:    memc [ 2 ].y [ 1 ] = 1.0;
153:    memc [ 2 ].x [ 2 ] = 20.0;
154:    memc [ 2 ].y [ 2 ] = 1.0;
155:
156: /*
157:    fuz_graph ( 5 , 5 , mema [ 0 ] );
158:    fuz_graph ( 5 , 130 , mema [ 1 ] );
159:    fuz_graph ( 5 , 255 , memb [ 0 ] );
160:    fuz_graph ( 210 , 5 , memb [ 1 ] );
161:    fuz_graph ( 210 ,130 , memc [ 0 ] );
162:    fuz_graph ( 210 , 255 , memc [ 1 ] );
163:    fuz_graph ( 415 , 5 , memc [ 2 ] );
164: */
165:
166:    /*      規則登録
167:            if x is A1 and y is B1 then z is C1.
168:            if x is A1 and y is B2 then z is C2.
169:            if x is A2 and y is B1 then z is C1.
170:            if x is A2 and y is B2 then z is C3.
171:    */
172:    num = fuz_rule ( &(mema [ 0 ]) , &(memb [ 0 ]) , &(memc [ 0 ]));
173:    num = fuz_rule ( &(mema [ 0 ]) , &(memb [ 1 ]) , &(memc [ 1 ]));
174:    num = fuz_rule ( &(mema [ 1 ]) , &(memb [ 0 ]) , &(memc [ 2 ]));
175:    num = fuz_rule ( &(mema [ 1 ]) , &(memb [ 1 ]) , &(memc [ 0 ]));
176:
177:    /*
178:    -------------------------------------
179:            Graphic mode 768x512
180:            16 colors 1 page
181:                            for X68000
182:    -------------------------------------
183:    */
184:
185:    screen ( 2 , 0 , 1 , 1 );
186:    apage ( 0 );
187:    vpage ( 1 );            /* グラフィック画面表示 */
188:    home ( 0 , 0 , 0 );
189:
190:    l = -1;
191:    vs = -1;
192:    v = -1;
193:    while ( -1 ){
194:            printf ( "¥n*****************初期値設定(終了はctrl+c)¥n" );
195:            do {
196:                    printf ( "¥n車間距離(0-40m):" );
197:                    scanf ( "%lf" , &l );
198:            } while (( l < 0 ) || ( l > 40 ));
199:            printf ( "l:%lf¥n" , l );
200:            do {
201:                    printf ( "先頭車速度 (0-70km ):" );
202:                    scanf ( "%lf" , &vs );
203:            } while (( vs < 0 ) || ( vs > 70 ));
204:            printf ( "vs:%lf¥n" , vs );
205:            do {
206:                    printf ( "後続車(ファジィ制御)速度 (0-70km ):" );
207:                    scanf ( "%lf" , &v );
208:            } while  (( v < 0 ) || ( v > 70 ));
209:            printf ( "v:%lf¥n" , v );
210:            printf ( "時間の単位 ( 0～10秒：だいたい1秒 ) :" );
211:            scanf ( "%lf" , &t );
212:            if (( t < 0 ) || ( t > 10 )){
213:                    t = 1.0;
214:            }
215:            t /= 3600;
216:            printf ( "t:%lf(h)¥n" , t );
217:            /*      車線を描く      */
218:            wipe ();
219:            line ( 490 , 0 , 490 , 511 , 15 , 0xffff );
220:            line ( 520 , 0 , 520 , 511 , 15 , 0xff00 );
221:            line ( 550 , 0 , 550 , 511 , 15 , 0xffff );
222:            simu ( l , v , vs , t );
223:
224:    }
225: }
226:
```

リスト2 FUZ_MEM.C

```
  1: /*
  2: ***************************************************************************
  3:     FUZzy_MEMbership function
  4:     ファジィ・メンバーシップ関数
  5:
  6:     台形型のメンバーシップ関数を構造体で
  7:     表現する。
  8:                     Programmed by T.Noma
  9:
 10:            ver 1.0         '90.10.13
 11:            ver 1.1         OR関係のバグとり
 12:            ver 1.2         アルゴリズムの見直し、修正
 13:            ver 1.3         グラフで目盛り、タイトル表示
 14:            ver 1.4         またバグとり。もういやっこんな生活!
 15:                            パッチあててなんとかなったかな。
 16:            ver 1.5         まだ、なんともならない。なんとかなった。
 17:            ver 1.6         範囲を超えても大丈夫にする。
 18:                            fuz_val関数を変更。
 19:            ver 1.7         エラーチェックの強化
 20: ***************************************************************************
 21: */
 22: #include <stdio.h>
 23: #include <stdlib.h>
 24: #include <string.h>
 25: #include <graph.h>
 26: #include "fuz.h"
 27:
 28: /*
 29: -------------グローバル変数-------------
 30:     だいたいファジィ・ルールベースを
 31:     意味するのだ。最大値はRULE_MAX
 32: ---------------------------------------
 33: */
 34: extern      int     rule_num ;
 35: extern      struct  MEMBER_TAG rule_x [ RULE_MAX ] , rule_y [ RULE_MAX ] ,
rule_z [ RULE_MAX ];
 36:
 37: /*
 38: -------------------------------------
 39:     FUZZYのNOTをとる
 40:     帰り値はなんと構造体のポインタ
 41:     をやめて、無難にポインタ渡しに
 42:     しましょう。
 43:     *temp = not ( * mem )
 44: -------------------------------------
 45: */
 46: void        fuz_not ( struct MEMBER_TAG *temp , struct MEMBER_TAG *mem )
 47: {
 48:     int     i , n = 0;
 49:
 50:     temp->min = mem->min;
 51:     temp->max = mem->max;
 52:
 53:     /* Xが最小値であるかを調べる */
 54:     if ( mem->min < mem->x [ 0 ] ){
 55:             temp->x [ 0 ] = mem->min;
 56:             temp->y [ 0 ] = 1.0;
 57:             n = 1;
 58:     }
 59:     for ( i = 0 ; i < mem->num ; i ++ ){
 60:             temp->x [ n ] = mem->x [ i ];
 61:             temp->y [ n ] = 1.0 - mem->y [ i ];
 62:             n ++;
 63:     }
 64:     if ( mem->max > mem->x [ mem->num - 1 ] ){
 65:             temp->x [ n ] = mem->max;
 66:             temp->y [ n ] = 1.0;
 67:             n ++;
 68:     }
 69:     temp->num = n;
 70: }
 71:
 72: /*
 73: ---------------------------------------
 74:     Superior 最大値をとる
 75:     帰り値は最大値、ポインタでその場所
 76: ---------------------------------------
 77: */
 78: double      fuz_sup ( struct MEMBER_TAG mem , double *address )
 79: {
 80:     int     i;
 81:     double  max = 0.0;
 82:
 83:     for ( i = 0 ; i < mem.num ; i ++ ){
 84:             if ( max < mem.y [ i ] ){
 85:                     max = mem.y [ i ];
 86:                     *address = mem.x [ i ];
 87:                     if ( max == 1.0 ){
 88:                             break;
 89:                     }
 90:             }
 91:     }
 92:     return max;
 93: }
 94:
 95: /*
 96: ---------------------------------------
 97:     Inferior 最小値をとる
 98:     帰り値は最小値、ポインタでその場所
 99: ---------------------------------------
100: */
101: double      fuz_inf ( struct MEMBER_TAG mem , double *address )
```


▶最近は寒さが身にしみませんか？ 私の部屋はもうシベリアも顔負けですが，そんな中にずっといるX68000君はなんだか元気がないみたいです。カゼをひいたとかで……。カゼには十分に注意しましょう。
太田 哲雄(16)北海道

```
102: {
103:     int      i;
104:     double  min = 1.0;
105:
106:     for ( i = 0 ; i < mem.num ; i ++ ){
107:             if ( min > mem.y [ i ] ){
108:                     min = mem.y [ i ];
109:                     *address = mem.x [ i ];
110:                     if ( min == 0.0 ){
111:                             break;
112:                     }
113:             }
114:     }
115:     return min;
116: }
117:
118: /*
119: ----------------------------------------
120:     FUZ_VAL
121:     Xの場の値を求める。
122:     帰り値は勿論DOUBLE型
123:     最大値を超えたら最後の値を返す。
124:     同様に最小値より小さければ、最小の
125:     時の値を返す。
126: ----------------------------------------
127: */
128: double      fuz_val ( double x , struct MEMBER_TAG mem )
129: {
130:     int      i ;
131:     double  x1 , x2 , y1 , y2 , y;
132:
133:     if ( x < mem.x [ 0 ] ){
134:             return ( mem.y [ 0 ] );
135:     }
136:     for ( i = 1 ; i < mem.num ; i ++ ){
137:             if ( x <= mem.x [ i ] ){
138:                     x1 = mem.x [ i - 1 ];
139:                     y1 = mem.y [ i - 1 ];
140:                     x2 = mem.x [ i ];
141:                     y2 = mem.y [ i ];
142:                     if (( x2 - x1 ) == 0 ){
143:                             return ( y2 );
144:                     }
145:                     else if (( x - x1 ) == 0 ){
146:                             return ( y1 );
147:                     }
148:                     y = ( y2 - y1 ) / ( x2 - x1 ) * ( x - x1 ) + y1;
149:                     return ( y );
150:             }
151:     }
152:     return ( mem.y [ mem.num - 1 ] );
153: }
154:
155: /*
156: ---------------------------------------------------------------
157:     FUZ_CROSS
158:     二線間の交点を求める。
159:     条件：必ず交わっていること。ya1 > yb1 and ya2 < yb2 など
160:     帰り値はdouble型のyの値。その時のXはポインタ指定。
161: ---------------------------------------------------------------
162: */
163: double      fuz_cross ( double xa1 , double ya1 , double xa2 , double ya2
164:              , double xb1 , double yb1 , double xb2 , double yb2 , double *
x )
165: {
166:     double  y;
167:     double  kata1 , kata2;  /* katamuki */
168:     double  dxa , dxb;      /* Xの偏差 */
169:
170:     dxa = xa2 - xa1;
171:     dxb = xb2 - xb1;
172:     if ( dxa == 0 ){
173:             *x = xa1;
174:             kata1 = 1e308;
175:     }
176:     else kata1 = ( ya2 - ya1 ) / dxa;
177:     if ( dxb == 0 ){
178:             *x = xb1;
179:             kata2 = 1e308;
180:     }
181:     else kata2 = ( yb2 - yb1 ) / dxb;
182:     if (( dxa == 0 ) && ( dxb != 0 )){
183:             return ( yb1 + ( yb2 - yb1 ) /  dxb * ( xa2 - xa1 ));
184:     }
185:     if (( dxa != 0 ) && ( dxb == 0 )){
186:             return ( ya1 + ( ya2 - ya1 ) /  dxa * ( xb2 - xb1 ));
187:     }
188:
189:     if ( kata1 == kata2 ){
190:             *x = xa2;
191:             return ( ya2 );
192:     }
193:     *x = ( yb1 - ya1 + kata1 * xa1 - kata2 * xb1 ) / ( kata1 - kata2);
194:     y = kata1 * ( *x - xa1 ) + ya1;
195:     if (( y < -0.0001 ) || ( y > 1.0 )){
196:             printf ( "xa1:%lf  xa2:%lf   ya1:%lf  ya2:%lf ¥n" , xa1 , xa2 ,
 ya1 , ya2 );
197:             printf ( "xb1:%lf  xb2:%lf   yb1:%lf  yb2:%lf ¥n" , xb1 , xb2 ,
 yb1 , yb2 );
198:             printf ( "y  :%lf¥n" , y );
199:             exit ( 0 );
200:     }
201:     return ( y );
202: }
203:
204:
205: /*
206: ----------------------------------------------
207:     FUZzy_GRAPH
208:     メンバーシップ関数をグラフにする
209:     大胆な関数。いいねえ。
210:
211:     xs , ys : 描画する左上の座標
212:     mem     : メンバーシップ関数の構造体
213: ----------------------------------------------
214: */
215: void        fuz_graph ( int xs , int ys , struct MEMBER_TAG mem )
216: {
217:     double  i;
218:     int     j;
219:     int     xmax = 200;
220:     int     ymax = 120;
221:     double  k;              /* 倍率 */
222:     int     x1 , y1 , x0 , y0;
223:
224:     if ( mem.max == mem.min ){
225:             printf ( "¥n最大値と最小値が同じです。¥n" );
226:             return ;
227:     }
228:     k = ( double ) xmax / ( mem.max - mem.min ) ;
229:     fill ( xs , ys , xs + xmax , ys + ymax , 2 );
230:     for ( i = 0 ; i <= xmax ;  i += k  ){
231:             line (( int )i +xs , ys + 20 , (int)i + xs , ys + ymax , 3 , 0x
aaaa );
232:     }
233:     for ( i = 20 ; i <= ymax ; i += 10 ){
234:             line ( xs , (int)i + ys , xs + xmax , (int)i + ys , 3 , 0xaaaa
);
235:     }
236:     for ( i = 0 ; i <= xmax ;  i += k * 10  ){
237:             line ( (int)i + xs , ys + 20 , (int)i + xs , ys + ymax , 10 , 0
xffff );
238:     }
239:     line ( xs , ys + 70 , xs + xmax , ys + 70 , 10 , 0xffff );
240:     line ( xs , ys + ymax , xs + xmax , ys + ymax , 15 , 0xffff );
241:     line ( xs , ys , xs , ys + ymax , 15 , 0xffff );
242:     symbol ( xs + 8 , ys + 2 , mem.name , 1 , 1 , 1 , 9 , 0 );
243:
244:     x0 = ( int )(( mem.x [ 0 ] - mem.min ) * k );
245:     y0 = 0;
246:     for ( j = 0 ; j < mem.num ; j ++ ){
247:             x1 = ( int )(( mem.x [ j ] - mem.min ) * k );
248:             y1 = ( int )( - 100 * mem.y [ j ] );
249:             line ( x0 + xs , y0 + ys + ymax , x1 + xs , y1 + ys + ymax , 13
, 0xffff );
250:             x0 = x1;
251:             y0 = y1;
252:     }
253: }
254:
255: /*
256: ----------------------------------------------
257:     FUZ_SAVE
258:     ファジィ・メンバーシップ関数のセーブ
259:
260:     char      *fname  : ファイルネーム
261:     struct .. mem     : メンバーシップ関数構造体
262:     return       1    : OK
263:                  0    : ERR
264: ----------------------------------------------
265: */
266: int         fuz_save ( char *fname , struct MEMBER_TAG *mem )
267: {
268:     FILE    *fp;
269:     int     i;
270:
271:     if (( fp = fopen ( fname , "w" )) == NULL ){
272:             printf ( "%s をopen出来ませんでした。¥n" , fname );
273:             return ( ERR );
274:     }
275:     printf ( "%sのファイルネームでセーブ中!¥n" , fname );
276:
277:     fprintf ( fp , "%s¥n" , mem->name );
278:     fprintf ( fp , "%5d%22.17lf%22.17lf¥n" , mem->num , mem->min , mem->max
);
279:
280:     for ( i = 0 ; i < mem->num ; i ++ ){
281:             fprintf ( fp , "%22.17lf%22.17lf¥n" , mem->x [ i ] , mem->y [ i
] );
282:     }
283:
284:     if ( fclose ( fp ) == EOF ){
285:             printf ( "正常にclose出来ませんでした。¥n" );
286:             return ( ERR );
287:     }
288:     printf ( "close完了¥n" );
289:     return ( OK );
290: }
291:
292: /*
293: ----------------------------------------------
294:     FUZ_LOAD
295:     ファジィ・メンバーシップ関数のロード
296:
297:     char      *fname  : ファイルネーム
298:     struct .. mem     : メンバーシップ関数構造体
299:     return       1    : OK
300:                  0    : ERR
301: ----------------------------------------------
302: */
303: int         fuz_load ( char *fname , struct MEMBER_TAG *mem )
304: {
```

▶危険物取扱試験を受けにいきました。なんと，その日は朝から雪で，なんか不安だったんですが，結構簡単でした。しかし，結果が3月の中頃とはなんとも遅い……。
松本　勝正(17)兵庫県

```
305:    FILE    *fp;
306:    int     i;
307:    int     num;
308:    double  min , max;
309:    double  x , y;
310:
311:    if (( fp = fopen ( fname , "r" )) == NULL ){
312:            printf ( "%s をopen出来ませんでした。¥n" , fname );
313:            return ( ERR );
314:    }
315:    printf ( "%sのファイルネームでロード中!¥n" , fname );
316:
317:    fgets ( mem->name , 32 , fp );
318:    fscanf ( fp , "%5d%lf%lf¥n" ,  &num , &min , &max );
319:    mem->num = num;
320:    mem->min = min;
321:    mem->max = max;
322:
323:    printf ( "num:%d¥n" , mem->num );
324:
325:    for ( i = 0 ; i < mem->num ; i ++ ){
326:            fscanf ( fp , "%lf%lf¥n" , &x, &y );
327:            mem->x [ i ] = x;
328:            mem->y [ i ] = y;
329:    }
330:
331:    if ( fclose ( fp ) == EOF ){
332:            printf ( "正常にclose出来ませんでした。¥n" );
333:            return ( ERR );
334:    }
335:    printf ( "close完了¥n" );
336:    return ( OK );
337: }
338:
339: /*
340: -------------------------------------
341:    FUZ_CPY
342:    メンバーシップ関数のコピー
343: -------------------------------------
344: */
345: void        fuz_cpy ( struct MEMBER_TAG *temp , struct MEMBER_TAG *mem )
346: {
347:    int     i;
348:
349:    strncpy ( temp->name , mem->name , CHARBUF );
350:    temp->num = mem->num;
351:    temp->min = mem->min;
352:    temp->max = mem->max;
353:    for ( i = 0 ; i < mem->num ; i ++ ){
354:            temp->x [ i ] = mem-> x [ i ];
355:            temp->y [ i ] = mem-> y [ i ];
356:    }
357: }
358:
```

```
359: /*
360: -------------------------------------
361:    FUZ_TRI
362:    三角型のメンバーシップ関数を
363:    つくる優れ物。
364:    centerを中心として2 * rangeが底辺の
365:    二等辺三角形を作る。
366: -------------------------------------
367: */
368: void        fuz_tri ( struct MEMBER_TAG *temp , double center , double rang
e
369:             , double val , double min , double max )
370: {
371:    double  x0 , x1 , x2 , y0 = 0 , y1 , y2 = 0;    /* for calculation */
372:    x1 = center;
373:    if (( val <= 1.0 ) && ( val >= 0.0 )){
374:            y1 = val;
375:    }
376:    else {
377:            printf ( "¥n値が可笑しい。val in fuz_tri¥n" );
378:            exit ( 0 );
379:    }
380:    if (( center < min ) || ( center > max )){
381:            printf ( "¥n値が可笑しい。center in fuz_tri¥n" );
382:            exit ( 0 );
383:    }
384:    if ( range == 0 ){
385:            printf ( "¥n底辺( range )が0じゃあーりませんか。困ります。¥n" );
386:            return ;
387:    }
388:    x0 = center - range ;
389:    x2 = center + range ;
390:    if ( x0 < min ){
391:            y0 = val / range * ( min - x0 );
392:            x0 = min;
393:    }
394:    if ( x2 > max ){
395:            y2 = val / range * ( x2 - max );
396:            x2 = max;
397:    }
398:
399:    temp->num = 3;
400:    temp->min = min;
401:    temp->max = max;
402:    temp->x [ 0 ] = x0;
403:    temp->x [ 1 ] = x1;
404:    temp->x [ 2 ] = x2;
405:    temp->y [ 0 ] = y0;
406:    temp->y [ 1 ] = y1;
407:    temp->y [ 2 ] = y2;
408: }
409:
410:
```

リスト3 FUZ_ENG.C

```
  1: /*
  2: **********************************************************************
  3:    FUZzy_estimating_ENGine
  4:    ファジィ推論エンジン
  5:
  6:    台形型のメンバーシップ関数を構造体で
  7:    表現したものを、ルールに基づき
  8:    推論する。
  9:                    Programmed by T.Noma
 10:
 11:            ver 1.0         '90.10.17
 12:            ver 1.1         言語ヘッジ関数追加
 13: **********************************************************************
 14: */
 15: #include <stdio.h>
 16: #include <math.h>
 17: #include "fuz.h"
 18:
 19: /*
 20: ------------グローバル変数------------
 21:    だいたいファジィ・ルールベースを
 22:    意味するのだ。最大値はRULE_MAX
 23: ----------------------------------------
 24: */
 25: extern     int     rule_num ;
 26: extern     struct  MEMBER_TAG rule_x [ RULE_MAX ] , rule_y [ RULE_MAX ] ,
rule_z [ RULE_MAX ];
 27:
 28: /*
 29: ---------------------------------------
 30:    FUZ_AREA
 31:    囲まれた一区間の面積を求める
 32: ---------------------------------------
 33: */
 34: double     fuz_area ( double x1 , double y1 , double x2 , double y2 )
 35: {
 36:    double  men;
 37:
 38:    men = ( x2 - x1 ) * ( y1 + y2 ) / 2.0;
 39:    return ( men );
 40: }
 41:
 42: /*
 43: ---------------------------------------
 44:    FUZ_GRAVity
 45:    重心を得る
 46: ---------------------------------------
```

```
 47: */
 48: double     fuz_grav ( struct MEMBER_TAG mem)
 49: {
 50:    int     i , j ;
 51:    double  m [ DMAX ];                 /* 区間毎の面積 */
 52:    double  mf = 0.0 , ml = 0.0 ;   /* 前者の面積  、  後者の面積 */
 53:    int     mfi , mli;                  /* 比較用の整数型変数。1000倍して比較 */
 54:    double  xg;
 55:    double  b , c , d;
 56:    double  k , kk;
 57:
 58:    for ( i = 1 ; i < mem.num ; i ++ ){
 59:            m [ i ] = fuz_area ( mem.x [ i - 1 ] , mem.y [ i - 1 ] ,
 60:                                 mem.x [ i ] , mem.y [ i ] );
 61:    }
 62:
 63:    for ( i = 1 ; i < mem.num ; i ++ ){
 64:            mf = 0.0;
 65:            ml = 0.0;
 66:            for ( j = 1 ; j <= i ; j ++ ){
 67:                    mf += m [ j ];
 68:            }
 69:            for ( j = i + 1 ; j < mem.num ; j ++ ){
 70:                    ml += m [ j ];
 71:            }
 72:            mfi = ( int ) 1000 * mf;
 73:            mli = ( int ) 1000 * ml;
 74:            if ( mfi > mli ){
 75:                    break;
 76:            }
 77:            else if ( mfi == mli ){
 78:                    /* なんてきりがいいんでしょう、丁度境界線だよ */
 79:                    return ( mem.x [ i ] );
 80:            }
 81:    }
 82:    k = ( mem.y [ i - 1 ] - mem.y [ i ] ) / ( mem.x [ i ] - mem.x [ i - 1 ]
);
 83:    b = -2 * k * mem.x [ i ] - 2 * mem.y [ i ];
 84:    c = - ( mf - ml )  + mem.x [ i ] * ( k * mem.x [ i ] + 2 * mem.y [ i ]
);
 85:    d = b * b - 4.0 * k * c;
 86: /*
 87:    printf  ( "D= %lf¥n" , d );
 88: */
 89:    if (( k <= 0.0000001 ) && ( k > -0.0000001 )){
 90:            xg = -c / b;
```


▶先日，札幌雪まつりに行ってきました。寒いうえにすごい人込みだったけど，雪像がキレイで，とてもていねいに作ってあることに感心してしまいました。鈴木亜久里とオグリキャップを合体させた「鈴木オグリ」という馬がヘルメットをかぶった雪像はとてもインパクトがあった。
中鉢　進一(22)北海道

```
91:     }
92:     else {
93:             xg = ( - b + sqrt ( d )) / k / 2.0;
94:             if (( xg > mem.x [ i ] ) || ( xg < mem.x [ i - 1 ] )) {
95:                     xg = ( - b - sqrt ( d )) / k / 2.0;
96:             }
97:     }
98:     return ( xg );
99: }
100:
101: /*
102: ----------------------------------------
103:     FUZ_ACUT
104:     αカットしたものをtempにいれる
105: ----------------------------------------
106: */
107: void  fuz_acut ( struct MEMBER_TAG *temp , struct MEMBER_TAG *mem  , doubl
e val )
108: {
109:     struct MEMBER_TAG alfa;
110:
111:     alfa.num = 2;
112:     alfa.min = mem->min;
113:     alfa.max = mem->max;
114:     alfa.x [ 0 ] = mem->min;
115:     alfa.y [ 0 ] = val;
116:     alfa.x [ 1 ] = mem->max;
117:     alfa.y [ 1 ] = val;
118:     if ( val != 0 ){
119:             fuz_and ( temp , mem , &alfa );
120:     }
121:     else if ( val == 1.0 ){
122:             fuz_cpy ( temp , mem );
123:     }
124:     else if ( val == 0.0 ){
125:             fuz_cpy ( temp , &alfa );
126:     }
127:     else printf ( "そんなことはないはず。fuz_acut¥n" );
128: }
129:
130: /*
131: ----------------------------------------
132:     言語ヘッジに利用される演算・・・らしい
133:     FUZ_SCL         スカラー倍
134:     FUZ_NORM        正規化
135:     FUZ_BEKI        べき乗
136:     FUZ_CON         集中化
137:     FUZ_DIL         拡大化
138: ----------------------------------------
139: */
140: void       fuz_scl ( double k , struct MEMBER_TAG *mem )
141: {
142:     int        i ;
143:     double  kk , *dummy;
144:
145:     kk = 1.0 / fuz_sup ( *mem , dummy );
146:     /*      最大はあくまでも1なのだ */
147:     if ( k > kk ) k = kk;
148:     for ( i = 0 ; i < mem->num ; i ++ ){
149:             mem->y [ i ] *= k;
150:     }
151:
152: }
153: void       fuz_norm ( struct MEMBER_TAG *mem )
154: {
155:     double  k , dummy;
156:     double  sup;
157:
158:     sup = fuz_sup ( *mem , &dummy );
159:     k = 1.0 / sup;
160:     fuz_scl ( k , mem );
161: }
162: void       fuz_beki ( double k , struct MEMBER_TAG *mem )
163: {
164:     int     i;
165:
166:     for ( i = 0 ; i < mem->num ; i ++ ){
167:             mem->y [ i ] = pow ( mem->y [ i ] , k );
168:     }
169: }
170: void       fuz_con ( struct MEMBER_TAG *mem )
171: {
172:     fuz_beki ( 2.0 , mem );
173: }
174: void       fuz_dil ( struct MEMBER_TAG *mem )
175: {
176:     fuz_beki ( 0.50 , mem );
177: }
```

リスト4 FUZ_RULE.C

```
1: /*
2: ************************************************************************
3:     FUZzy_RULE_base
4:     ファジィルールベース
5:
6:     台形型のメンバーシップ関数を構造体で
7:     表現したものを、ルールに登録する。
8:     またそのデータベースに値を渡すことで
9:     自動的に最終結果が帰ってくると
10:     嬉しい。
11:                     Programmed by T.Noma
12:
13:             ver 1.0         '90.10.21       F1 Grand Prix in SUZUKA
14:                                             亜久里3位おめでとう!!
15:                                             中島入賞おめでとう!!
16:             ver 2.0         推論エンジンを二つにする。一方は途中経過
17:                             を表示する。
18: ************************************************************************
19: */
20: #include <stdio.h>
21: #include <math.h>
22: #include <string.h>
23: #include "fuz.h"
24:
25: /*
26: -------------グローバル変数-------------
27:     だいたいファジィ・ルールベースを
28:     意味するのだ。最大値はRULE_MAX
29: ----------------------------------------
30: */
31: extern      int      rule_num ;
32: extern      struct  MEMBER_TAG rule_x [ RULE_MAX ] , rule_y [ RULE_MAX ] ,
rule_z [ RULE_MAX ];
33:
34: /*
35: ----------------------------------------
36:     FUZ_RULE
37:     ファジイ・ルールベース登録
38:     帰り値はルールベース登録番号
39: ----------------------------------------
40: */
41: int         fuz_rule ( memx , memy , memz )
42: struct      MEMBER_TAG *memx , *memy , *memz;
43: {
44:
45:     fuz_cpy ( &rule_x [ rule_num ] , memx );
46:     fuz_cpy ( &rule_y [ rule_num ] , memy );
47:     fuz_cpy ( &rule_z [ rule_num ] , memz );
48:
49:     rule_num ++;
50:     return ( rule_num );
51: }
52:
53: /*
54: ----------------------------------------
55:     FUZ_MIN , FUZ_MAX
56:     その名の通りに二つの引き数の小さい方、
57:     大きい方を返す関数
58: ----------------------------------------
59: */
60: double      fuz_min ( double a , double b )
61: {
62:     if ( a < b ) return ( a );
63:     else return ( b );
64: }
65: double      fuz_max ( double a , double b )
66: {
67:     if ( a < b ) return ( b );
68:     else return ( a );
69: }
70:
71: /*
72: ----------------------------------------
73:     FUZ_ESTimate &
74:     FUZ_ESTimate_with_Graphics
75:     ファジィ推論  (実数にて演算)
76:     及び途中経過のグラフ付
77:
78:     ルールへの値をもらって、
79:     最終的な推論結果を出す
80:     夢の推論マシーン!!・・・
81:     なわけないか
82:     帰り値は勿論推論結果のdouble型
83: ----------------------------------------
84: */
85: double      fuz_est_g ( double a , double b , int xs , int ys )
86: {
87:     int     i;
88:     double  val_x ;
89:     double  val_y ;
90:     double  val_z ;
91:     static  struct  MEMBER_TAG mem_za [ RULE_MAX ] , temp1 , temp2;
92:     double  ret;
93:     char    fname [ 30 ];
94:
95:     /*
96:             if x(i) is A(i) and if y(i) is B(i) then z(i) is C(i)
97:
98:             もし規則XがA ( x軸:a ) で、規則YがB( b )であるとき、
99:             その各々の値 (Y軸:val_a , val_b) をとりそのAND (min:val_z) をとる。
100:             そしてそのmin値によって規則Zをαカットする (mem_za [ i ])。
101:            それを総ての規則群に行い、規則Zの結果のOR (最後のtemp1 ) をとる。
102:            それに対して重心法を用い、結果の帰り値とする。
103:     */
104:     for ( i = 0 ; i < rule_num ; i ++ ){
105:             val_x = fuz_val ( a , rule_x [ i ] );
106:             val_y = fuz_val ( b , rule_y [ i ] );
107:             val_z = fuz_min ( val_x , val_y );
108:             fuz_acut ( &mem_za [ i ] , &rule_z [ i ] , val_z );
109:             fuz_graph ( xs , ys , mem_za [ i ] );
110:             getch ();
111:     }
112:
```

▶献血のCMで近鉄の野茂が,「高校時代, 友達と献血しました」といっているのを聞くと,
「野茂の血をくれ」と思ってしまうのは僕だけでしょうか?　林 泰裕(19)石川県

```
113:     fuz_cpy ( &temp1 , &mem_za [ 0 ] );
114:     for ( i = 1 ; i < rule_num ; i ++ ){
115:             fuz_graph ( xs , ys , temp1 );
116:             fuz_or ( &temp2 , &mem_za [ i ] , &temp1 );
117:             fuz_cpy ( &temp1 , &temp2 );
118:             getch();
119:     }
120:     strcpy ( temp1.name , "演算結果" );
121:     fuz_graph ( xs , ys , temp1 );
122:     fuz_save ( "temp1" , &temp1 );
123:     getch();
124:     ret = fuz_grav ( temp1 );
125:     return  ( ret );
126: }
127:
128: double      fuz_est ( double a , double b )
129: {
130:     int      i;
131:     double   val_x ;
132:     double   val_y ;
133:     double   val_z ;
134:     static   struct  MEMBER_TAG mem_za [ RULE_MAX ] , temp1 , temp2;
135:     double   ret;
136:     char     fname [ 30 ];
137:
138:     /*
139:     **********************************************************************
140:             if x(i) is A(i) and if y(i) is B(i) then z(i) is C(i)
141:
142:             もし規則XがA ( x軸:a ) で、規則YがB( b )であるとき、
143:             その各々の値 ( Y軸:val_a , val_b) をとりそのAND (min:val_z) をとる。
144:               そしてそのmin値によって規則Zをαカットする (mem_za [ i ]) 。
145:             それを総ての規則群に行い、規則Zの結果のOR (最後のtemp1 ) をとる。
146:             それに対して重心法を用い、結果の帰り値とする。
147:     ***********************************************************************
148:     */
149:     for ( i = 0 ; i < rule_num ; i ++ ){
150:             val_x = fuz_val ( a , rule_x [ i ] );
151:             val_y = fuz_val ( b , rule_y [ i ] );
152:             val_z = fuz_min ( val_x , val_y );
153:             fuz_acut ( &mem_za [ i ] , &rule_z [ i ] , val_z );
154:     }
155:
156:     fuz_cpy ( &temp1 , &mem_za [ 0 ] );
157:     for ( i = 1 ; i < rule_num ; i ++ ){
158:             fuz_or ( &temp2 , &mem_za [ i ] , &temp1 );
159:             fuz_cpy ( &temp1 , &temp2 );
160:     }
161:     strcpy ( temp1.name , "演算結果" );
162:     ret = fuz_grav ( temp1 );
163:     return  ( ret );
164: }
165:
166:
```

リスト5 FUZ_OPR.C

```
  1: /*
  2: ********************************************************************
  3:     FUZzy_membership_Operating function
  4:     ファジィ・メンバーシップ操作関数
  5:
  6:     and とorだけ専門。New Algorithm
  7:                     Programmed by T.Noma
  8:
  9:             ver 1.0         '90.10.13
 10:             ver 1.1         x_sortをor,and関数の中に入れる
 11:             ver 1.2         cross判定アルゴリズムの再構成
 12:             ver 1.3         crossn関数の変更→返り値をフラグにする
 13:             ver 1.4         更にcross判定を変更
 14:             ver 1.5         ifdef文を入れる。デバッグはDBGを定義
 15:             ver 1.6         crossn関数のエラーチェック強化
 16:             ver 1.7         等しいかをみる関数equalを追加
 17:             ver 1.8         アルゴリズム変更
 18: ********************************************************************
 19: */
 20: #include <stdio.h>
 21: #include <stdlib.h>
 22: #include <string.h>
 23: #include "fuz.h"
 24:
 25: #define     SEIDO   1e6
 26: #define     EQ      0
 27: #define     GT      1
 28: #define     LT      -1
 29: #define     ON      1
 30: #define OFF         0
 31: /*
 32: -------------グローバル変数-------------
 33:     だいたいファジィ・ルールベースを
 34:     意味するのだ。最大値はRULE_MAX
 35: ----------------------------------------
 36: */
 37: extern      int      rule_num ;
 38: extern      Mem rule_x [ RULE_MAX ] , rule_y [ RULE_MAX ] , rule_z [ RULE_M
AX ];
 39:
 40: /*
 41: --------------------------------------
 42:     等しいかを調べる
 43:     精度はSEIDOにて定義
 44:     a < b = -1
 45:     a > b =  1
 46:     a == b = 0
 47: --------------------------------------
 48: */
 49: static      int      equal ( double a , double b )
 50: {
 51:     int      aa , bb;
 52:     aa = ( int )( a * SEIDO );
 53:     bb = ( int )( b * SEIDO );
 54: #ifdef      DBG
 55:     printf ( "eqaul関数  a:%d   b:%d ¥n" , a , b );
 56: #endif
 57:     if (( aa + 1 ) < bb ) return  LT;
 58:     else if (( aa - 1 ) > bb ) return GT;
 59:     else return EQ;
 60: }
 61:
 62: /*
 63: --------------------------------------
 64:     まず横方向に大きいかどうかを
 65:     調べながらソートする。
 66:     返り値はスタックに積んだ数
 67: --------------------------------------
 68: */
 69: static      int      x_sort ( Stack stk [ ] , Mem *mema , Mem *memb )
 70: {
 71:     int      a_cnt = 0 , b_cnt = 0 , s_cnt = 0;
 72:     int      i;
 73:     double  xa , xb , ya , yb;
 74:
 75:     /*      比べる物の範囲をチェック            */
 76:     if (( mema->min != memb->min ) && ( mema->max != memb->max )){
 77:             printf ( "警告!比較する構造体の範囲が違ってます。¥n a->min:%lf b->min:
%lf" ,
 78:                     mema->min , memb->min );
 79:             printf ( " a->max:%lf b->max:%lf¥n" , mema->max , memb->max );
 80:     }
 81: #ifdef DBG
 82:     printf ( "x_sort¥n" );
 83: #endif
 84:     /*      スタックの中をすべてクリア          */
 85:     for ( i = 0 ; i < SMAX ; i ++ ){
 86:             stk [ i ].upper = ' ';
 87:             stk [ i ].same [ 0 ] = ' ';
 88:             stk [ i ].same [ 1 ] = ' ';
 89:             stk [ i ].lower = ' ';
 90:     }
 91:     while (( a_cnt <  mema->num ) && ( b_cnt < memb->num )){
 92:             xa = mema->x [ a_cnt ];
 93:             xb = memb->x [ b_cnt ];
 94: #ifdef DBG
 95:             printf ( "a:%d  b:%d  s:%d xa:%lf xb:%lf¥n" , a_cnt , b_cnt , s
_cnt ,
 96:                                                             xa , xb );
 97: #endif
 98:             switch ( equal ( xa , xb )){
 99:             case LT:
100:                     ya = mema->y [ a_cnt ];
101:                     yb = fuz_val ( xa , *memb );
102:                     push_stk ( &( stk [ s_cnt ] ) , ya , yb , 'a' , ' ' );
103:                     a_cnt ++;
104:                     break;
105:             case GT:
106:                     ya = fuz_val ( xb , *mema );
107:                     yb = memb->y [ b_cnt ];
108:                     push_stk ( &( stk [ s_cnt ] ) , yb , ya , 'b' , ' ' );
109:                     b_cnt ++;
110:                     break;
111:             case EQ:
112:                     ya = mema->y [ a_cnt ];
113:                     yb = memb->y [ b_cnt ];
114:                     /*      同じでなければa , bの両方を入れてあげる */
115:                     if ( push_stk ( &( stk [ s_cnt ] ) , ya , yb , 'a' , 'b
' ) != SAME ){
116:                             push_stk ( &( stk [ s_cnt ] ) , yb , ya , 'b' ,
'a' );
117:                     }
118:                     a_cnt ++;
119:                     b_cnt ++;
120:                     break;
121:             default:
122:                     break;
123:             }
124:             s_cnt ++;
125:     }
126:     /*      残った物の方が当然でかいと解釈してセット        */
127:     for ( i = a_cnt ; i < mema->num ; i ++ ){
128:             stk [ s_cnt ].upper = 'a';
129:             s_cnt ++;
130:     }
131:     for ( i = b_cnt ; i < memb->num ; i ++ ){
132:             stk [ s_cnt ].upper = 'b';
133:             s_cnt ++;
134:     }
135: #ifdef DBG
136:     /*      スタックの中をすべて見せる          */
137:     for ( i = 0 ; i < s_cnt ; i ++ ){
138:             printf ( "ussl:%c:%c:%c:%c:¥n" ,
139:             stk [ i ].upper ,
140:             stk [ i ].same [ 0 ] ,
141:             stk [ i ].same [ 1 ] ,
142:             stk [ i ].lower );
143:     }
```

▶もうすぐ,アマチュア無線の国家試験。しかし,期末テストが重なってしまった。落ちたら先生に不幸の手紙を書こう……。　渡辺　一史(17)愛知県

```
144: #endif
145:    return ( s_cnt );
146: }
147:
148: /*
149: ------------------------------------------------------------
150:    Stackに入れる作業
151:
152:    y1が大きければ0。同じならば1。小さいと2が返り値
153: ------------------------------------------------------------
154: */
155: static      int      push_stk ( Stack *stk , double y1 , double y2 , unsigne
d char mark , unsigned char emark )
156: {
157: #ifdef DBG
158:    printf ( "y1:%lf  y2:%lf  mark:%c:emark:%c:¥n" , y1 , y2 , mark , emark
);
159:    printf ( "y1 == y2 :%d¥n" , y1 == y2 );
160: #endif
161:    switch ( equal ( y1 , y2 )){
162:    case EQ:
163:            stk->same [ 0 ] = mark;
164:            stk->same [ 1 ] = emark;         /* End Mark */
165:            return ( SAME );
166:            break;
167:    case GT:
168:            stk->upper = mark;
169:            return ( UPPER );
170:            break;
171:    case LT:
172:            stk->lower = mark;
173:            return ( LOWER );
174:            break;
175:    default:
176:            printf ( "そんな筈はない。...push_stk¥n" );
177:            break;
178:    }
179: }
180:
181: /*
182: ------------------------------------------------------------------
183:    FUZ_CROSS_New
184:    二線間の交点を求める。
185:    条件：必ず交わっていること。ya1 > yb1 and ya2 < yb2 など
186:    帰り値はOK,ERR。double型のx，yはポインタ指定。
187: ------------------------------------------------------------------
188: */
189: int          fuz_crossn ( double *x , double *y , Mem *mema , Mem *memb , in
t a_cnt , int b_cnt )
190: {
191:    double xa1 , ya1 , xa2 , ya2;
192:    double xb1 , yb1 , xb2 , yb2;
193:    double kata1 , kata2;  /* katamuki */
194:    double dxa , dxb;       /* Xの偏差 */
195:
196:    xa1 = mema->x [ a_cnt - 1 ];
197:    ya1 = mema->y [ a_cnt - 1 ];
198:    xa2 = mema->x [ a_cnt ];
199:    ya2 = mema->y [ a_cnt ];
200:    xb1 = memb->x [ b_cnt - 1 ];
201:    yb1 = memb->y [ b_cnt - 1 ];
202:    xb2 = memb->x [ b_cnt ];
203:    yb2 = memb->y [ b_cnt ];
204:
205:    dxa = xa2 - xa1;
206:    dxb = xb2 - xb1;
207:    if ( dxa == 0 ){
208:            *x = xa1;
209:            kata1 = 1e308;
210:    }
211:    else kata1 = ( ya2 - ya1 ) / dxa;
212:    if ( dxb == 0 ){
213:            *x = xb1;
214:            kata2 = 1e308;
215:    }
216:    else kata2 = ( yb2 - yb1 ) / dxb;
217:
218:    if (( dxa == 0 ) && ( dxb != 0 )){
219:            *y = ( yb1 + ( yb2 - yb1 ) /  dxb * ( xa2 - xa1 ));
220:            return OK;
221:    }
222:    else if (( dxa != 0 ) && ( dxb == 0 )){
223:            *y = ( ya1 + ( ya2 - ya1 ) /  dxa * ( xb2 - xb1 ));
224:            return OK;
225:    }
226:    else if (( dxa == 0 ) && ( dxb == 0 )){
227: #ifdef DBG
228:            printf ( "わっかんないんだよー。dxa == dxb == 0 ...fuz_crossn¥n" );
229: #endif
230:            return ERR;
231:    }
232:
233:    if ( kata1 == kata2 ){
234: #ifdef DBG
235:            printf ( "傾きが同じだぞ...fuz_crossn¥n" );
236: #endif
237:            return ERR;
238:    }
239:    *x = ( yb1 - ya1 + kata1 * xa1 - kata2 * xb1 ) / ( kata1 - kata2);
240:    *y = kata1 * ( *x - xa1 ) + ya1;
241:    if (( *y < -0.0 ) || ( *y > 1.0 )){
242: #ifdef DBG
243:            printf ( "値がおかしいですね。... fuz_crossn¥n" );
244:            printf ( "xa1:%lf  xa2:%lf   ya1:%lf  ya2:%lf ¥n" , xa1 , xa2 ,
ya1 , ya2 );
245:            printf ( "xb1:%lf  xb2:%lf   yb1:%lf  yb2:%lf ¥n" , xb1 , xb2 ,
yb1 , yb2 );
246:            printf ( "y  :%lf  x  :%lf¥n" , *y , *x);
247: #endif
248:            return ERR;
249:    }
250:    if ((( *x < xa1 ) || ( *x > xa2 )) && (( *x < xb1) || ( *x > xb2 ))){
251: #ifdef DBG
252:            printf ( "値がおおきんじゃない？...fuz_crossn¥n" );
253:            printf ( "xa1:%lf  xa2:%lf   ya1:%lf  ya2:%lf ¥n" , xa1 , xa2 ,
ya1 , ya2 );
254:            printf ( "xb1:%lf  xb2:%lf   yb1:%lf  yb2:%lf ¥n" , xb1 , xb2 ,
yb1 , yb2 );
255:            printf ( "y  :%lf  x  :%lf¥n" , *y , *x);
256: #endif
257:            return ERR;
258:    }
259:    return ( OK );
260: }
261: /*
262: ----------------------------------------
263:    Set_Member
264:    x,yをセットする
265: ----------------------------------------
266: */
267: void        set_mem ( Mem *temp , int *t_cnt , Mem *mem , int cnt )
268: {
269:    if ( *t_cnt > 0 ){
270:            if ( temp->x [ *t_cnt - 1 ] < mem->x [ cnt ] ){
271:                    temp->x [ *t_cnt ] = mem->x [ cnt ];
272:                    temp->y [ *t_cnt ] = mem->y [ cnt ];
273:                    ++ ( *t_cnt );
274:            }
275:            else if (( temp->x [ *t_cnt - 1 ] == mem->x [ cnt ] ) &&
276:                    ( temp->y [ *t_cnt - 1 ] != mem->y [ cnt ] )){
277:                    temp->x [ *t_cnt ] = mem->x [ cnt ];
278:                    temp->y [ *t_cnt ] = mem->y [ cnt ];
279:                    ++ ( *t_cnt );
280:            }
281:    }
282:    else {
283:            temp->x [ *t_cnt ] = mem->x [ cnt ];
284:            temp->y [ *t_cnt ] = mem->y [ cnt ];
285:            ++ ( *t_cnt );
286:    }
287: }
288: /*
289: ----------------------------------------
290:    Set_Member_by_xy
291:    x,yをセットする
292: ----------------------------------------
293: */
294: static      int      set_xy ( Mem *temp , int *t_cnt , double x , double y )
295: {
296:    if ( *t_cnt > 0 ){
297:            if ( temp->x [ *t_cnt - 1 ] < x ){
298:                    temp->x [ *t_cnt ] = x;
299:                    temp->y [ *t_cnt ] = y;
300:                    ++ ( *t_cnt );
301:                    return OK;
302:            }
303:            else if (( temp->x [ *t_cnt - 1 ] == x ) &&
304:                    ( temp->y [ *t_cnt - 1 ] != y )){
305:                    temp->x [ *t_cnt ] = x;
306:                    temp->y [ *t_cnt ] = y;
307:                    ++ ( *t_cnt );
308:                    return OK;
309:            }
310:            return ERR;
311:    }
312:    else {
313:            temp->x [ *t_cnt ] = x ;
314:            temp->y [ *t_cnt ] = y ;
315:            ++ ( *t_cnt );
316:    }
317:    return OK;
318: }
319: /*
320: -------------------------------------------------------
321:    FUZ_OR New Version
322:
323:    memaとmembをStack stkに従って
324:    Stackから取り出す。この時新たな交点も調べる。
325:    結果はtempに入れる。
326: -------------------------------------------------------
327: */
328: void        fuz_or ( Mem *temp , Mem *mema , Mem *memb )
329: {
330:    unsigned char   last , now , buf [ 2 ] , low , lastlow;
331:    unsigned char   a_cond , b_cond , lsetp;                    /* a ,
bの後の状態 */
332:    int     a_cnt = 0 , b_cnt = 0 , s_cnt = 0 , t_cnt = 0;
333:    int     i , sw = 0;
334:    double  xc , yc;        /* cross point  */
335:    double  xa , xb;
336:    static  Stack   stk [ SMAX ];
337:    int     s_max;
338:
339:    if (( mema->min != memb->min ) || ( mema->max != memb->max )){
340:            printf ( "¥n比べる対象の範囲が違ってます ( func fuz_orn )¥n" );
341:            return ;
342:    }
343:    temp->min = mema->min;
344:    temp->max = mema->max;
345:
346:    /*      ソートする      */
347:    s_max = x_sort ( stk , mema , memb );
348:
349:    xa = mema->x [ a_cnt ];
```

▶花粉症のはびこるこの時期，編集部の方々は健全でいらっしゃるでしょうか？　今年は昨年よりも花粉の量が多いんですって。　佐久間　崇(17)栃木県

```
350:    xb = memb->x [ b_cnt ];
351:    lsetp = 's'; /* 一番最初にsがきてもセットするように初期設定した */
352:    while ( s_cnt < s_max ){
353:            now = stk [ s_cnt ].upper;
354:            buf [ 0 ] = stk [ s_cnt ].same [ 0 ];
355:            buf [ 1 ] = stk [ s_cnt ].same [ 1 ];
356:            low = stk [ s_cnt ].lower;
357: #ifdef DBG
358:            printf ( "now:%c:t_cnt:%d  a_cnt:%d  b_cnt:%d¥n" , now , t_cnt
, a_cnt , b_cnt );
359: #endif
360:            /*      前と違うということは・・・！そう、交点があるのだ。      */
361:            if ((( now == 'a' ) && ( a_cond == 'l' )) ||
362:                (( low == 'a' ) && ( a_cond == 'u' ))){
363: #ifdef DBG
364:                    printf ( "fuz_crossnを呼ぶ¥n" );
365: #endif
366:                    if ( fuz_crossn ( &xc , &yc , mema , memb , a_cnt , b_c
nt ) == OK ){
367:                            if ( set_xy ( temp , &t_cnt , xc , yc ) == OK )
{
368:                                    /* 状態をリセットする */
369:                                    a_cond = ' ';
370:                                    b_cond = ' ';
371:                            }
372:                    }
373:            }
374:            if ((( now == 'b' ) && ( b_cond == 'l' )) ||
375:                (( low == 'b' ) && ( b_cond == 'u' ))){
376: #ifdef DBG
377:                    printf ( "fuz_crossnを呼ぶ¥n" );
378: #endif
379:                    if ( fuz_crossn ( &xc , &yc , mema , memb , a_cnt , b_c
nt ) == OK ){
380:                            if ( set_xy ( temp , &t_cnt , xc , yc ) == OK )
{
381:                                    /* 状態をリセットする */
382:                                    a_cond = ' ';
383:                                    b_cond = ' ';
384:                            }
385:                    }
386:            }
387:            /*      現在の点をセット        */
388:            switch ( now ){
389:            case 'a' :
390:                    set_mem ( temp , &t_cnt , mema , a_cnt ++ );
391:                    a_cond = 'u';
392:                    lsetp = 'a';
393:                    break;
394:            case 'b' :
395:                    set_mem ( temp , &t_cnt , memb , b_cnt ++ );
396:                    b_cond = 'u';
397:                    lsetp = 'b';
398:                    break;
399:            /*      upperがないとき、それ以外のカウンタを進める。   */
400:            case ' ' :
401:                    sw = OFF;
402:                    for ( i = 0 ; i < 2 ; i ++ ){
403:                            if ( buf [ i ] == lsetp ){
404:                                    /*      lsetpと同じ場合はsameでもセットする
 */
405:                                    switch ( lsetp ){
406:                                    case 'a' :
407:                                            set_mem ( temp , &t_cnt , mema
, a_cnt );
408:                                            break;
409:                                    case 'b' :
410:                                            set_mem ( temp , &t_cnt , memb
, b_cnt );
411:                                            break;
412:                                    default :
413:                                            break;
414:                                    }
415:                                    sw = ON;
416:                                    break;
417:                            }
418:                            /* 最初がもしsameでも大丈夫、セットする */
419:                            else if ( lsetp == 's' ){
420:                                    switch ( buf [ i ] ){
421:                                    case 'a' :
422:                                            set_mem ( temp , &t_cnt , mema
, a_cnt );
423:                                            break;
424:                                    case 'b' :
425:                                            set_mem ( temp , &t_cnt , memb
, b_cnt );
426:                                            break;
427:                                    default :
428:                                            break;
429:                                    }
430:                                    sw = ON;
431:                                    break;
432:                            }
433:                    }
434:                    /*      カウンタを進める        */
435:                    for ( i = 0 ; i < 2 ; i ++ ){
436:                            if ( buf [ i ] == 'a' ){
437:                                    a_cnt ++;
438:                                    if ( sw == ON ) a_cond = 's';
439:                            }
440:                            else if ( buf [ i ] == 'b' ){
441:                                    b_cnt ++;
442:                                    if ( sw == ON ) b_cond = 's';
443:                            }
444:                    }
445:                    break;
446:            default :
447:                    printf ( "fuz_orn... デフォルトになってしまった。:%c:¥n" , n
ow );
448:                    break;
449:            }
450:            if ( low == 'a' ){
451:                    a_cnt ++;
452:                    a_cond = 'l';
453:            }
454:            else if ( low == 'b' ){
455:                    b_cnt ++;
456:                    b_cond = 'l';
457:            }
458:            s_cnt ++;
459:    }
460:    temp->num = t_cnt;
461: #ifdef DBG
462:    fuz_save ( "con" , temp );
463: #endif
464: }
465:
466: /*
467: -------------------------------------------------------
468:    FUZ_AND New Version
469:
470:    memaとmembをStack stkに従って
471:    Stackから取り出す。この時新たな交点も調べる。
472:    結果はtempに入れる。
473: -------------------------------------------------------
474: */
475: void        fuz_and ( Mem *temp , Mem *mema , Mem *memb )
476: {
477:    unsigned char   last , now , buf [ 2 ] , upper , lastup;        /* 最後
の点 */
478:    unsigned char   a_cond , b_cond , lsetp;                        /* a ,
bの後の状態 */
479:    int     a_cnt = 0 , b_cnt = 0 , s_cnt = 0 , t_cnt = 0;
480:    int     i , sw;
481:    double  xc , yc;        /* cross point  */
482:    double  xa , xb;
483:    static  Stack   stk [ SMAX ];
484:    int     s_max;
485:
486:    if (( mema->min != memb->min ) || ( mema->max != memb->max )){
487:            printf ( "¥n比べる対象の範囲が違ってます ( func fuz_orn )¥n" );
488:            return ;
489:    }
490:    temp->min = mema->min;
491:    temp->max = mema->max;
492:
493:    /*      ソートする      */
494:    s_max = x_sort ( stk , mema , memb );
495:
496:    xa = mema->x [ a_cnt ];
497:    xb = memb->x [ b_cnt ];
498:    lsetp = 's';
499: #ifdef DBG
500:    printf ( "s_cnt:%d¥n" , s_cnt );
501: #endif
502:    while ( s_cnt < s_max ){
503:            now = stk [ s_cnt ].lower;      /*      and     */
504:            buf [ 0 ] = stk [ s_cnt ].same [ 0 ];
505:            buf [ 1 ] = stk [ s_cnt ].same [ 1 ];
506:            upper = stk [ s_cnt ].upper;
507: #ifdef DBG
508:            printf ( "now:%c:t_cnt:%d  a_cnt:%d  b_cnt:%d¥n" , now , t_cnt
, a_cnt , b_cnt );
509: #endif
510:            /*      前と違うということは・・・！そう、交点があるのだ。      */
511:            if ((( now == 'a' ) && ( a_cond == 'u' )) ||
512:                (( upper == 'a' ) && ( a_cond == 'l' ))){
513: #ifdef DBG
514:                    printf ( "fuz_crossnを呼ぶ¥n" );
515: #endif
516:                    if ( fuz_crossn ( &xc , &yc , mema , memb , a_cnt , b_c
nt ) == OK ){
517:                            temp->x [ t_cnt ] = xc;
518:                            temp->y [ t_cnt ] = yc;
519:                            t_cnt ++;
520:                            /* 状態をリセットする */
521:                            a_cond = ' ';
522:                            b_cond = ' ';
523:                    }
524:            }
525:            if ((( now == 'b' ) && ( b_cond == 'u' )) ||
526:                (( upper == 'b' ) && ( b_cond == 'l' ))){
527: #ifdef DBG
528:                    printf ( "fuz_crossnを呼ぶ¥n" );
529: #endif
530:                    if ( fuz_crossn ( &xc , &yc , mema , memb , a_cnt , b_c
nt ) == OK ){
531:                            temp->x [ t_cnt ] = xc;
532:                            temp->y [ t_cnt ] = yc;
533:                            t_cnt ++;
534:                            /* 状態をリセットする */
535:                            a_cond = ' ';
536:                            b_cond = ' ';
537:                    }
538:            }
539:            /*      現在の点をセット        */
540:            switch ( now ){
541:            case 'a' :
542:                    set_mem ( temp , &t_cnt , mema , a_cnt ++ );
543:                    a_cond = 'l';
544:                    lsetp = 'a';
545:                    break;
546:            case 'b' :
547:                    set_mem ( temp , &t_cnt , memb , b_cnt ++ );
548:                    b_cond = 'l';
```


▶とうとう，プリンタなるものを買ってしまった。しかし，熱転写式でランニングコストが高いので，おいそれとは使えない。まあ，安かったからしかたないか。

加藤　勲(20)埼玉県

```
549:                    lsetp = 'b';
550:                    break;
551:            /*      upperがないとき、それ以外のカウンタを進める。    */
552:            case ' ' :
553:                    sw = OFF;
554:                    for ( i = 0 ; i < 2 ; i ++ ){
555:                            if ( buf [ i ] == lsetp ){
556:                                    /*      lsetpと同じ場合はsameでもセットする
    */
557:                                    switch ( lsetp ){
558:                                    case 'a' :
559:                                            set_mem ( temp , &t_cnt , mema
, a_cnt );
560:                                            break;
561:                                    case 'b' :
562:                                            set_mem ( temp , &t_cnt , memb
, b_cnt );
563:                                            break;
564:                                    default :
565:                                            break;
566:                                    }
567:                                    sw = ON;
568:                                    break;
569:                            }
570:                            /* 最初がもしsameでも大丈夫、セットする */
571:                            else if ( lsetp == 's' ){
572:                                    switch ( buf [ i ] ){
573:                                    case 'a' :
574:                                            set_mem ( temp , &t_cnt , mema
, a_cnt );
575:                                            break;
576:                                    case 'b' :
577:                                            set_mem ( temp , &t_cnt , memb
, b_cnt );
578:                                            break;
579:                                    default :
580:                                            break;
581:                                    }
582:                                    sw = ON;
583:                                    break;
584:                            }
585:                    }
586:                    /*      カウンタを進める        */
587:                    for ( i = 0 ; i < 2 ; i ++ ){
588:                            if ( buf [ i ] == 'a' ){
589:                                    a_cnt ++;
590:                                    if ( sw == ON ) a_cond = 's';
591:                            }
592:                            else if ( buf [ i ] == 'b' ){
593:                                    b_cnt ++;
594:                                    if ( sw == ON ) b_cond = 's';
595:                            }
596:                    }
597:                    break;
598:            default :
599:                    printf ( "fuz_orn... デフォルトになってしまった。:%c:¥n" , n
ow );
600:                    break;
601:            }
602:            /*      and     */
603:            if ( upper == 'a' ){
604:                    a_cond = 'u';
605:                    a_cnt ++;
606:            }
607:            else if ( upper == 'b' ){
608:                    b_cond = 'u';
609:                    b_cnt ++;
610:            }
611:            s_cnt ++;
612:            if ( a_cnt >= mema->num ) break;
613:            if ( b_cnt >= memb->num ) break;
614:    }
615:    temp->num = t_cnt;
616: #ifdef DBG
617:    fuz_save ( "con" , temp );
618: #endif
619: }
620:
```

リスト6 FUZ.H

```
 1: /*
 2: ****************************************************
 3:    FUZzy_membership Header
 4:    ファジィ・メンバーシップ関数のヘッダ
 5:                    Programmed by T.Noma
 6:
 7:            ver 1.0         '90.10.15
 8:            ver 1.1         FUZ_ENG included
 9:            ver 1.2         FUZ_OPER included
10:            ver 1.3         fuz_crossn変更
11:            ver 1.4         FUZ_OPRに変更
12: ****************************************************
13: */
14:
15: #define CHARBUF         32              /* キャラクタの最大バッファ */
16: #define DMAX            30              /* データの最大数 */
17: #define ERR             0
18: #define NG              0
19: #define OK              1
20: #define RULE_MAX    10          /* 定義出来る規則の最大数 */
21: #define UPPER           0
22: #define SAME            1
23: #define LOWER           2
24: #define SMAX            60              /* 横軸の点の最大数(x_sort) */
25:
26: struct          MEMBER_TAG {
27:    unsigned char   name [ CHARBUF ];       /* メンバーシップ関数の名前 */
28:    int             num;                    /* データの数 */
29:    double          min;                    /* Xの最小値 */
30:    double          max;                    /* Xの最大値 */
31:    double          x [ DMAX ];             /* X座標の値 */
32:    double          y [ DMAX ];             /* Y座標の値 (0~1) */
33: };
34: typedef         struct  MEMBER_TAG      Mem;
35: typedef         struct  {
36:    unsigned char   upper;
37:    unsigned char   same [ 2 ];
38:    unsigned char   lower;
39: } Stack;
40:
41: /*
42: -------------プロトタイプ宣言-------------
43: */
44: void            fuz_not ( struct MEMBER_TAG *temp , struct MEMBER_TAG *mem );
45: double          fuz_sup ( struct MEMBER_TAG mem , double *address );
46: double          fuz_inf ( struct MEMBER_TAG mem , double *address );
47: double          fuz_val ( double x , struct MEMBER_TAG mem );
48: double          fuz_cross ( double xa1 , double ya1 , double xa2 , double ya2
49:                 , double xb1 , double yb1 , double xb2 , double yb2 , double *
x );
50: int             fuz_crossn ( double *x , double *y , Mem *mema , Mem *memb , in
t a_cnt , int b_cnt );
51: void            fuz_or ( Mem *temp , Mem *mema , Mem *memb );
52: void            fuz_and ( Mem *temp , Mem *mema , Mem *memb );
53: void            fuz_graph ( int xs , int ys , struct MEMBER_TAG mem );
54: int             fuz_save ( unsigned char *fname , struct MEMBER_TAG *mem );
55: int             fuz_load ( unsigned char *fname , struct MEMBER_TAG *mem );
56: void            fuz_cpy ( struct MEMBER_TAG *temp , struct MEMBER_TAG *mem );
57: void            fuz_tri ( struct MEMBER_TAG *temp , double center , double rang
e
58:                 , double val , double min , double max );
59: double          fuz_area ( double x1 , double y1 , double x2 , double y2 );
60: double          fuz_grav ( struct MEMBER_TAG mem);
61: void            fuz_acut ( struct MEMBER_TAG *temp , struct MEMBER_TAG *mem ,
double val );
62: void            fuz_scl ( double k , struct MEMBER_TAG *mem );
63: void            fuz_norm ( struct MEMBER_TAG *mem );
64: void            fuz_beki ( double k , struct MEMBER_TAG *mem );
65: void            fuz_con ( struct MEMBER_TAG *mem );
66: void            fuz_dil ( struct MEMBER_TAG *mem );
67:
68: int             fuz_rule ( struct MEMBER_TAG *memx , struct MEMBER_TAG *memy ,
struct MEMBER_TAG *memz );
69: double          fuz_min ( double a , double b );
70: double          fuz_max ( double a , double b );
71: double          fuz_est ( double a , double b );
72: double          fuz_est_g ( double a , double b , int x , int y );
```

リスト7 MK

```
 1: CC = cc
 2: CFLAGS = /Fc
 3: LC = cc
 4: LFLAGS = /W
 5:
 6: PROG0 = spdx.x
 7: COBJ0 = $(PROG0:x=o)
 8: CFILE0 = $(PROG0:x=c)
 9:
10: COBJ1 = fuz_mem.o
11: CFILE1 = $(COBJ1:o=c)
12:
13: COBJ2 = fuz_rule.o
14: CFILE2 = $(COBJ2:o=c)
15:
16: COBJ3 = fuz_eng.o
17: CFILE3 = $(COBJ3:o=c)
18:
19: COBJ4 = fuz_opr.o
20: CFILE4 = $(COBJ4:o=c)
21:
22: $(PROG0): $(COBJ0) $(COBJ1) $(COBJ2) $(COBJ3) $(COBJ4)
23:     $(LC) $(LFLAGS) $(COBJ0) $(COBJ1) $(COBJ2) $(COBJ3) $(COBJ4)
24: $(COBJ0): $(CFILE0)
25:     $(CC) $(CFLAGS) $(CFILE0)
26: $(COBJ1): $(CFILE1)
27:     $(CC) $(CFLAGS) $(CFILE1)
28: $(COBJ2): $(CFILE2)
29:     $(CC) $(CFLAGS) $(CFILE2)
30: $(COBJ3): $(CFILE3)
31:     $(CC) $(CFLAGS) $(CFILE3)
32: $(COBJ4): $(CFILE4)
33:     $(CC) $(CFLAGS) $(CFILE4)
34:
```

▶ついに買ったニューマシン「ユーノス・ロードスター」！　スピードよりもフィーリングを重視して開発されたところが気に入って……。「MX-5・ミアータ」のときからの憧れだったんだよなあ(36回ローン……)。　丸藤　俊之(22)神奈川県

ようこそここへC言語

文字列って何だろう(第6回)

Nakamori Akira
中森 章

今回はC言語のプログラミングを理解するのに重要なテーマのひとつ「文字列の扱い」を取り上げてみましょう。C言語による文字列操作はコンピュータ内部でのデータの扱われ方を反映したもので，それが自由度の高さとなっています。

「私をスキーに連れてって」を観てスキーに興味を持って以来，年を経ること数年，先日やっと生まれて初めてのスキーを体験した中森章です。予想どおり次の日は体中が痛くて会社を休んでしまい，スキーを始めるなら若いうちにしとけばよかったと思いっきり後悔してしまいました。

さて，今回は文字列を取り上げます。文字列の操作はC言語のもっとも得意とするところです。C言語はよくシステム記述言語といわれることがありますが，その本質は文字列を自由に扱えることにあるといっても過言ではありません。前回，前々回で説明した配列がプログラミング一般の基本だとすると，文字列の操作はC言語としてのプログラミングの基本になります。今回はこの文字列について学んでいくことにしましょう。

文字と文字列

プログラムは文字の集まりです。どのようなプログラミング言語で書いたプログラムでもそれを翻訳し実行するためにはいったんソースプログラムというテキストファイルの形式にしてやらなければなりません[1]。テキストファイルとはコンピュータが画面に文字として表示できるデータ（文字コード）のみを含んだファイルのことです。いかにコンピュータが高性能，高機能になろうとコンピュータのCPUが理解できるのは0と1が並んだ数値だけです。文字コードとはこの0と1をいくつかひとまとまりにしてできる数値を1対1に文字に対応させた約束ごとにすぎません。

図1
キャラクタコード表

上位4ビット→ / 下位4ビット↓

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	コントロールキャラクタ			0	@	P	'	p	2バイトコード文字の1バイト目			ー	タ	ミ	2バイトコード文字の1バイト目	
1			!	1	A	Q	a	q			。	ア	チ	ム		
2			"	2	B	R	b	r			「	イ	ツ	メ		
3			#	3	C	S	c	s			」	ウ	テ	モ		
4			$	4	D	T	d	t			、	エ	ト	ヤ		
5			%	5	E	U	e	u			・	オ	ナ	ユ		
6			&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
7			'	7	G	W	g	w			ァ	キ	ヌ	ラ		
8			(	8	H	X	h	x			ィ	ク	ネ	リ		
9			)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
A			*	:	J	Z	j	z			ェ	コ	ハ	レ		
B			+	;	K	[	k	{			ォ	サ	ヒ	ロ		
C			,	<	L	¥	l	|			ャ	シ	フ	ワ		
D			−	=	M	]	m	}			ュ	ス	ヘ	ン		
E			.	>	N	^	n	￣			ョ	セ	ホ	゛		
F			/	?	O	_	o				ッ	ソ	マ	゜		

図1にX68000で使用されている文字コード（1バイトコードのみ）の一覧を示します。これは8ビットの数値を1文字に対応させる約束ごとで，各コンピュータで統一がとれるようにISO（International Organization for Standardization＝国際標準化機構）やANSI（American National Standars Institute＝アメリカ国内規格協会）で決められた世界的な標準が基になっています[2]。なお，1バイトでは表せないような漢字などの文字は2バイトを1文字に対応させる約束になっていて，X68000ではその1バイト目は1バイトコードの表の一部に割り当てられています[3]。

文字についてもう少し具体的な例で考えましょう。エディタなどを用いてソースプログラムを作成することはファイル中にプログラムを表す文字コード（数値）を書き入れていく作業と考えることもできます。たとえば，

X＝X＋3；

というプログラムを作ることは16進で，

78, 3D, 78, 2B, 33, 3B

という数値をファイルに書き込むことと同じです。逆に，このソースファイルの先頭から1バイト分のデータを読み込むと78_Hという数値を得ることができます。これを単なる数値データであるか文字（x）であるかの区別をすることは一般的には不可能です。その解釈はプログラムを書く人の判断に任せられます。一般的には，読み込むファイルがデータファイルなら数値データですし，ソースファイル（テキストファイル）ならば文字ということになります。

要するに文字とは数に特別な意味を持たせたものであることを理解してください。つまり，文字と数（整数）とはまったく同じものなのです。それなのにFORTRANやPASCALなど多くのプログラミング言語では違うデータ型として数と文字を厳格に区別して使用することに

なっています。一方，C言語では数と文字の間に区別はありません。プログラミング言語の観点からは一概にどちらが優れているかはいえませんが，この違いが後のち文字や文字列の扱いやすさを決定してくるのです。

さて，次は文字列です。文字列とは文字の列，すなわち文字の並んだものです。これはデータ構造という観点から見ると文字の一次元配列となります[4]。ただし，それが通常の一次元配列と同等のものであるかというとそうでもありません。

文字列の実現方法は大別すると2通りの方法があります。第1の方法は一次元配列の第1要素に文字列の長さ（文字列に含まれる文字の個数）を格納し，第2要素以降に文字列に含まれる文字を格納する方法です。第2の方法は文字列の終わりを示す文字を特別に定義しておき，一次元配列の第1要素から順に文字列に含まれる文字を格納したあとに終わりを示す文字を余分に格納するという方法です。どちらも通常の一次元配列に何らかの情報を付加して文字列とみなす方法です。図2にこれらの方法の概念図を示しておきましょう。

しかし，これらの実現方法は通常プログラマには見えてこないものです。プログラミング言語の文法で定められたように文字列を使っていれば，プログラマは文字列がどのような方法で実現されているか知らなくても，あとはコンパイラなりインタプリタの処理系が勝手にうまくやってくれるからです。ところが，C言語では文字列の構造を文法でしっかりと規定してあるのです（図2の(b)の方法)。したがって，C言語で文字列を扱うときはその構造を念頭に置いてプログラムを書かなくてはいけません。これも文字が整数と同一であることと同じくらいC言語では重要な概念（というか前提）です。

---

1) 昔，プログラムをカードにパンチしてカード読み取り機から読み込ませて実行していた時代では当てはまらないかもしれない。しかしその場合でもカード上の文字が文字コードに変換されて翻訳，実行されていたことに変わりない。
2) 図1はよくASCII(ANSI)コードと呼ばれるが，正確にはそれを拡張したJISコードである。
3) これはシフトJISコードの場合である。漢字を表すコードはJISでも数種類定められているが，日本のパソコンでもっとも多く採用されているのはこのシフトJISと呼ばれる文字コードである。X68000でもシフトJISコードが採用されている。
4) C言語やPASCALでは文字列を文字の一次元配列として宣言する。FORTRANでは文字と文字列に区別はなくすべてが文字列(CHARACTER型）として扱われる。

C言語での文字と文字列

それでは，C言語での文字と文字列について具体的に説明します。C言語には通常8ビット整数を表すchar型というデータ構造がありますが，それこそその名前のとおり文字を格納するために存在するデータ型と言えます[5]。また，すでにご存じのように，文字を格納するだけでなく，char型は符号付きで−128から+127，符号なしで0から+255の範囲の整数を表すデータとしても使用されます。本質的に同じものはわざわざ区別する必要がないというのがC言語の思想なのでしょう。

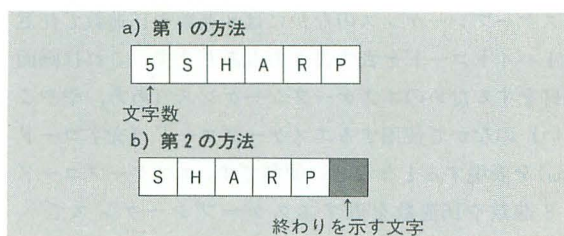



図2 文字列を文字の一次元配列で実行する方法

ところで，C言語では文字と数値を区別する必要はありませんが，プログラムの記述に便利なように，数値が文字であることを強調して示すための表記が用意されています。それは文字を「'」（シングルクォート＝単一引用符）で囲む表記です。すなわち'a'は61_Hという数値，'B'は42_Hという数値を文字として表現するために使用します（図1の文字コードを参照のこと)。これはC言語では文字定数と呼ばれています。プログラムの流れによっては，

```
if(x==0x42) { … };
```

と記述するよりも，

```
if(x=='B') { … };
```

と記述するほうが意味のはっきりする場合もあります（xには文字が格納されていることがわかる)。文字定数はこのような要求に応えるための機能なのです。また，文字定数はchar型整数と本質的に同じですから数値と文字を混合した，

```
x=ch-'a'+10;
```

などという表現も可能です。これは，chが'a'ならばxは10，chが'b'ならばxは11，chが'c'ならばxは12，……というように16進数を意味する'a'から'f'までの文字を10から15までの整数値に変換するための式になります[6]。

ところで，C言語ではキーボードなどで入力できないようなある種の文字（改行やタブ）を文字定数として表現するためにエスケープシーケンスが用意されています。ここでいうエスケープシーケンスはパソコンの画面を制御するときに使用されるエスケープシーケンスとはまったく関係ありません。C言語のエスケープシーケンスは＼（バックスラッシュ）のあとに文字をいくつか並べたものです。日本製のキーボードでは＼のキーがないことが多いのですが，その場合は＼と同じ文字コードを持つ￥という文字で代用されます。

図3にC言語のすべてのエスケープシーケンスを示しておきましょう。なお，図3に載っている以外の文字を＼のあとに書いたときはその文字自身を表します。たとえば，'w'と'＼w'は同じ文字になります(なんの意味があるのやら)。

エスケープシーケンスは一見は2文字とか3文字，4文字のようにも思えますが，それが意味するものはあくまでも1つの文字であることに注意してください。また，

エスケープシーケンスのなかには8進数や16進数で任意の1バイトコードを表す書き方もあります。これは画面制御をするためのエスケープシーケンス（ああ，ややこしい）のなかで使用するエスケープコード（文字コード$1B_H$)を表現するときなどに便利です。エスケープコードは8進数や16進数を表すエスケープシーケンスで'\033', '\x1b', '\x1B'などと表現することができます。

さて，次は文字列を説明しましょう。C言語では最終要素がヌル（値が0の文字，'\0'とも書く）である文字の一次元配列を文字列として定義します。たとえば，

abcdefg

という文字列を使用する場合，C言語では，

char a [ ] = {'a','b','c','d','e','f','g',0}；

というchar型の配列宣言をすればよいことになります。配列の大きさを宣言してない（第4回で説明した不完全な配列の宣言を参照のこと）のは文字列に含まれる文字数を数えるのが面倒だからです。本来なら文字列に含まれる文字数より1大きい（ヌル文字のため）値（か，それ以上の値）を配列の大きさとして書かなければならないところです。

ところで，文字に文字定数があったように文字列にも文字列定数（文字列リテラルともいう）があります。これは文字列を「"」(ダブルクォート＝二重引用符）で囲んで表します。上の文字列は，

"abcdefg"

というようにも記述できるのです。また，ヌル文字だけを含む文字列定数は，

" "

で表現されます。文字列は文字の配列ですから文字定数を示すためのエスケープシーケンスを文字列定数の中で使用してもかまいません。printf関数を使用するときの書式指定（これは文字列ですね）で，

"x=%d¥n"

といった表現を見ることがありますが，これは文字列の中に改行を示すエスケープシーケンスがある例ですね。

次に，文字列定数"abcdefg"は文字列を示す配列を初期化するときの値である，

{'a','b','c','d','e','f','g',0}

と構造的にまったく同じです。すなわち，文字列定数は初期化付きのchar型の一次元配列と同じものということができます。したがって文字列の宣言は，

char a [ ] = "abcdefg"；

という表現でも行うことができます（こっちのほうが一般的）。これは"abcdefg"という名無し（配列要素だけを指定している）の一次元配列に名前を与える作業と思ってもかまいません。

なお，C言語の文字列の宣言には，

char ＊a = "abcdefg"；

などという方法もあります。これはポインタを使用する場合です。ポインタについてはこの連載の先の回で説明する予定ですから，ここでは深入りしません。文字列定数の宣言にポインタを使用しても配列を使用しても実用上は大差ありませんから，とりあえず配列で覚えておくことにしましょう[7]。

図3 C言語のエスケープシーケンス

文字	値	意味
\a	07_H	ベル文字
\b	08_H	バックスペース
\f	$0C_H$	改頁
\n	$0A_H$	改行
\r	$0D_H$	復帰
\t	09_H	水平タブ
\v	$0B_H$	垂直タブ
\\	$5C_H$	バックスラッシュ
\?	$3F_H$	疑問符
\'	27_H	単一引用符
\"	22_H	二重引用符
\ooo		8進数
\xhh		16進数

ooo 1〜3桁の8進数
hh 1〜2桁の16進数

---

5) charとはcharacter（文字の意）の略。漢字を別にすればたいていの文字は8ビットで表すことができる。

6) この式は文字コード上で'a', 'b', 'c', 'd', 'e', 'f'が順番に1ずつ異なる値で並んでいることを前提としている。すべてのC言語の処理系が同じ文字コード体系を採用しているとは限らないので，プログラムの移植を考えるなら特定の文字コード体系に依存したプログラムを書くべきではない。しかし，ほとんどすべてのC言語の処理系はASCIIコード体系を採用しているので，ASCIIコード体系を前提としたプログラムは（K＆Rの中でさえ）よく見られる。

7) 実際のところ配列とポインタでは少し意味が違う。配列とポインタを混同することによりとんでもないエラーを発生することもある。詳しく知りたい人はK＆Rの第5章(とくに5.5 文字ポインタと関数）や『C言語を256倍使うための本』（アスキー出版）の「究極の添字」の章などを参照のこと。

◆基礎力を高めよう

設問1 次に示す文字列定数の内で文法的に正しいものには○を正しくないものには×をつけてください。

1)" "　　2)"¥0"
3)"abc¥0efg"　　4)" " "
5)" ' "　　6)" " " "
7)"¥0182"　　8)"¥0128"
9)"¥a¥b¥c¥d¥e¥f"　　10)"¥xyz"

設問2 文字列の長さ（strlen関数の値）とは文字列を示すchar型の一次元配列の要素（文字）の中でヌル文字に到達するまでの要素の個数として定義されます。設問1に示された文法的に正しい文字列定数の長さをすべて求めてください。

（解答は128ページ）

文字列を操作する関数

C言語は文字や文字列の扱いを得意とするプログラミング言語ですが文字列をそのまま扱う演算子を持っていません（文字に関しては整数と同じだから演算子はある

といえる)。BASICやFORTRANなどのプログラミング言語では文字列を連結するための演算子などを用意していて，こっちのほうが文字列を扱うのに便利そうに思えます。ところが，C言語ではあえて特別な演算子は用意せず，文字列の構造を文字（char型整数）の一次元配列として規定し，普通の配列と同じ手順で文字列を扱えるようにしています。この点，C言語では文字と整数の型変換や文字列型という特殊なデータ型を考える必要がありません。逆説めいていますが，文字列というデータ型を排除した（というか，配列で間に合わせた）という点がC言語を文字列の扱いを得意にさせている理由ではないでしょうか。

たとえば，

```
"This is a pen."
```

という文字列の最初の4文字を"That"という文字列に置き換えることを考えます。これがFORTRANなら，

```
CHARACTER A*14
```

と宣言したあと，

```
A='This is a pen.'
A(1:4)='That'
```

というように部分文字列を参照する演算子を用いるプログラムになるでしょう。これがC言語では，

```
char A[ ]="This is a pen." ;
```

という宣言のあとは，

```
A[0]='T';
A[1]='h';
A[2]='a';
A[3]='t';
```

という配列への単純な代入ですんでしまいます。文字列用の特別な演算子は何もありませんからその文法を思い出す必要もありません。

あるいは，ある文字列Aを別の文字列B（C言語では配列）にコピーする操作を考えましょう。確かにFORTRANでは，

```
B=A
```

という一文でできてしまいますが，C言語では，繰り返し制御構造を用いて，たとえば，

```
for(i=0;A[i] !=0;i=i+1)
            B[i]=A[i];
```

という複雑な文を書かなければなりません。C言語では配列同士の代入はできませんから，配列の各要素をひとつずつ代入することになるのです。この例では一見FORTRANのほうが有利に思えますね。

それでは，少し問題を変形して文字列中の小文字を大文字に変換してコピーすることを考えましょう。FORTRANでは文字列の中から1文字取り出しては小文字か否かを調べ，小文字ならそれを大文字に変換したあとに目的の文字列に代入していかなければなりません。たとえば，

```
DO 10 I=1 TO LEN(A)
C=A(I:I)
IF(LGE C.'a').AND. LLE(C,'z'))THEN
  C=CHAR(ICHAR(C)-ICHAR('a')+ICHAR('
A'))
    ENDIF
10  B(I:I)=C
```

というプログラムになるのでしょうか（正しいかどうか知らないよ)。少し条件が変わっただけでものすごいプログラムの変更量ですね。一方，C言語では，

```
for(i=0;A[i] !=0;i=i+1) {
   if(A[i]>='a' && A[i]<='z')
           B[i]=A[i]-'a'+'A' ;
   else
           B[i]=A[i];
}
```

となってそれほど大きな変更ではありません。この例ではFORTRANのプログラムのほうがかえって複雑になっています。これは一見便利そうに思える機能も標準的な使用方法をはずれたら無力になってしまう例といえるでしょう。

C言語では文字を整数と同等に扱えるということ（文字と数の大小比較や演算を行える）もプログラムの変更量が少ない理由になっています。このように，C言語では文字列の操作をすべて配列の各要素の操作に帰着してしまったことによって文字列操作に柔軟性を持たせているのです。

とはいえ，頻度の高い文字列の操作を毎回配列要素の操作として書き下すのは大変です。そこでC言語ではよく使う文字や文字列の操作をライブラリ関数として提供しています。図4にANSI規格で規定されている文字と文字列に関するライブラリ関数を示します（この中のtoupperという関数を使えば先の例題はもっと単純になりますね)。ライブラリで供給されてはいますが，もちろんこれらの関数はC言語のプログラムで簡単に書くこともできます[8]。ライブラリ関数をC言語で書き直すことは文字や文字列の操作を理解する練習になりますから関数の作り方を知っている人は挑戦してみましょう（関数の作り方は次回で説明する予定です)。

とりあえず，ここではライブラリ関数の中でよく使われると思われるstrcpy, strcat, strcmpの3つについて簡単に使い方を説明しておきましょう（それ以外はマニュアルを見てね)。

●strcpy

第1引数　文字列を示す配列名
　　　　　または文字列へのポインタ
第2引数　文字列を示す配列名
　　　　　または文字列定数
　　　　　または文字列へのポインタ

返り値　第1引数の文字列へのポインタ
機能　第2引数で示す文字列を第1引数で示す文字列にコピーする
例　char A [100], B [100] ;
char C []="Gundam F91";
strcpy(A,C);
strcpy(B,"Oh!X");
このあと，Aは"Gundam F91"，Bは"Oh!X"という文字列になる。

●strcat
第1引数　文字列を示す配列名
または文字列へのポインタ

図4
ANSI規格で定義されている文字，文字列関数

関数	機能
文字変換	
toupper	大文字に変換
tolower	小文字に変換
文字テスト	
isalpha	アルファベットか
isupper	大文字か
islower	小文字か
isdigit	数字か
isxdigit	16進数字か
isspace	空白か
ispunct	区切り文字か
isalnum	数字かアルファベットか
isprint	表示可能文字か(空白含む)
isgraph	表示可能文字か(空白以外)
iscntrl	制御文字か
文字列変換	
atof	文字列→double型　変換
atoi	文字列→int型　変換
atol	文字列→long型　変換
strtod	文字列→double型　変換
strtol	文字列→long型　変換
strtoul	文字列→unsigned long型　変換
文字列比較	
memcmp	領域を比較(ヌル文字無視)
strcmp	文字列を比較
strcoll*	文字コード表を指定して文字列を比較
strncmp	文字列の最初の数文字を比較
strxfrm*	文字コード表で文字列を変換
文字列連結	
strcat	文字列を連結
strncat	文字列の最初の数バイトを連結
文字列コピー	
memcpy	領域をコピー(ヌル文字無視)
memmove	重なりがある領域をコピー(ヌル文字無視)
strcpy	文字列をコピー
strncpy	文字列の最初の数文字をコピー
文字列関数	
memset	領域を指定文字で満たす
strerror	エラーコードに対応するメッセージを返す
strlen	文字列の長さを返す
文字列検索	
memchr	領域中に指定文字を探す
strchr	文字列中に指定文字を探す
strcspn	文字列中の指定文字以外をスキップする
strpbrk	文字列中に複数文字のどれかを探す
strrchr	文字列中に指定文字を逆方向に探す
strspn	文字列中の指定文字をスキップする
strstr	文字列中に指定文字列を探す
strtok	文字列をトークンに切り分ける

＊印の関数はXCのライブラリにない。

第2引数　文字列を示す配列名
または文字列定数
または文字列へのポインタ
返り値　第1引数の文字列へのポインタ
機能　第2引数で示す文字列を第1引数で示す文字列の後ろにコピーして2つの文字列を連結する。
例　char A [100]="Gundam F91";
char B [100]="　";
char C [ ]= "/0083";
strcat(A,C);
strcat(B,"Oh!X");
このあと，Aは"Gundam F91/0083"，Bは"Oh!X"という文字列になる。

●strcmp
第1引数　文字列を示す配列名
または文字列定数
または文字列へのポインタ
第2引数　文字列を示す配列名
または文字列定数
または文字列へのポインタ
返り値　第1引数の文字列と第2引数の文字列の内容が等しいとき整数0，
第1引数の文字列が第2引数の文字列より小さいとき負の整数値，
第1引数の文字列が第2引数の文字列より大きいとき正の整数値
機能　第1引数の文字列と第2引数の文字列に含まれる文字を先頭から1文字ずつ比較して不一致を見つけた場合，第1引数側の文字列に含まれる文字が第2引数側の文字列に含まれる文字よりも（文字コードが）小さい場合には負の値，逆の場合は正の値を返す(文字コードの差を返す処理系が多い)。不一致が見つからない場合は0を返す。
例　char A [100]="Gundam F90";
char B [100]="Gundam";
char C [100]="Gundam F91";
ここで，strcmp(A,B)の値は正，
strcmp(B,"Gundam")の値は0，
strcmp(A,C)の値は負
となる。

8)　実際，同じ機能を持つ関数をC言語で書けばいいので文字や文字列関係のライブラリ関数は必要ないかもしれない。strstrなどという関数がライブラリとして存在することを何人の人が知っているのだろう(同等の関数を自作した人のほうが多いはずだ)。しかし，実際のライブラリ関数はアセンブリ言語で記述されていることが多いので，ライブラリ関数にある関数はそちらを使うほうがプログラムを高速に実行できる。

文字や文字列の入出力

さて，文字と文字列が出てきたところでそれらを入出力する方法を説明しておきましょう。いままで入力はscanf関数，出力はprintf関数を使用してきましたが，これらの関数の書式指定に文字と文字列を加えます。書式の中で文字は%cで指定し，文字列は%sで指定するようになっています。このとき入出力のオマジナイは次のようになります。

scanf("%c",&x) → 1文字をchar型の変数xに読み込む。

scanf("%s",x) → 文字列をchar型の配列xに読み込む。

printf("%c",x) → 変数xの値を文字コードとみなして文字をプリントする。

printf("%s",x) → char型配列xに格納された文字列をプリントする。

scanfで文字列を配列に読み込むとき変数名の前に&が不要なことに注意してください。また，scanf関数での%sは空白，タブ，改行といった区切り文字で分けられた文字列を読み込むようになっています。

たとえば，

```
char x [100] ;
main( )
{
        scanf("%s",x) ;
        printf("%s¥n",x) ;
}
```

というプログラムを実行して，キーボードから，

This is a pen.

という文字を打ち込むとき，配列xに読み込まれるのは空白で区切られた"This"という文字列です。そのためprintf関数でプリントされるのは，

This

という文字列だけになります。もし配列xに，

"This is a pen."

が読み込まれることを期待していたら失敗に終わります。

このように文字列を入力するときのscanf関数は少々使いづらいものがあります。そこで標準入力（通常はキーボード）から1行を読み込むために使われるのがgetsという関数です。gets関数を使うためのオマジナイは次のようになっています。

gets(x) → char型配列xに標準入力から1行分の文字列を読み込む。通常は読み込んだ文字列へのポインタ（配列xの先頭アドレス）を返すが，ファイルの終了コード（$1A_H$=CTRL-Z）を読み込むと0を返す。

たとえば，

```
char x [100] ;
main( )
{
        gets(x)  ;
        printf("%s¥n",x)  ;
}
```

というプログラムはキーボードからの1行の入力をそのまま画面にプリントします。

とりあえず，以上で紹介したscanf, printf, getsという3種類の関数を用いていれば，差し当たってのところは作成するプログラムの入出力に不便はないと思われるでしょう。

文字列を扱うプログラム

文字列を扱うプログラムの例として整数を文字列に変換するプログラムを書きましょう。これは図4に示したatoi関数の逆操作です。たとえば，123という整数値が与えられると"123"という文字列を適当な配列内に格納するようなプログラムです。実はXCのライブラリにはちょうど同じ働きをするitoaという関数が用意されています（これを使えば一発）が，ここでは配列要素をいじくりながら泥臭いプログラムを書いてみましょう。方針としては次のようになります。

●整数が負の数なら，負の数であることを覚えておき，その絶対値を新たな整数値とする。

●整数値が0なら"0"を配列にコピーする。

●商が0になるまで与えられた整数を10で割り続ける。このときに出てくる余りが，それぞれ1の桁，10の位，100の位，1000の位，……，の値となる。それらの数値を文字に変換して逆順に格納し直せば求める文字列ができる。

●与えられた整数が負の数であるなら'−'という文字を文字列の先頭につける。順序を逆転する前なら文字列の最後につける。

一度作成した文字列の順序を逆転させなければならないところが少々難です（美しくない）が，この方針で作成したプログラムがリスト1です。文字列の順序を逆転する方法は制御構造をよく理解している人ならば簡単にわかるはずです。なお，リスト1では数を文字に変換するために，数に'0'という文字を加えるという操作をしています。これは数の範囲が0から9である（10で割った余りだから）ことを利用した小技です。

さて，リスト1は整数を10進数の文字列に変換するプログラムでした。今度は16進数の文字列に変換することを考えましょう。リスト1との違いは，

●10で割る代わりに16で割って商と余りを求めること。

●16で割った余りを16進数の文字に変換する場合，た

リスト1

```
 1: /*
 2:          整数→文字列変換プログラム
 3: */
 4: int      number;                     /* ここに整数を読み込む       */
 5: int      sign;                       /* 符号を覚える                */
 6: char     string[100];                /* 変換結果をここに格納する */
 7:
 8: main()
 9: {
10:     int  i,j;
11:     char save;
12:
13:     scanf("%d",&number);            /* 整数の読み込み */
14:     printf("整数=%d ",number);   /* 読み込んだ値をプリント */
15:
16:     sign=0;
17:     if(number<0){                   /* 符号の記憶と絶対値 */
18:         sign=1;
19:         number=-number;
20:     }
21:     if(number==0)   strcpy(string,"0");
22:     else {
23:         for(i=0;number!=0;i=i+1){
24:             string[i]=(number % 10)+'0';
25:                                     /* 10で割った余りを文字に */
26:             number=number / 10; /* 10で割った商    */
27:         }
28:         /* この時点で文字は逆順に並んでいる */
29:         if(sign==1){
30:             string[i]='-';          /* マイナス符号を追加 */
31:             i=i+1;
32:         }
33:         string[i]=0;                /* ヌル文字を入れて文字列に */
34:         j=i-1;                      /* 文字列の最後の文字の位置(ヌル文字以外)*/
35:         for(i=0;i<j;i=i+1){       /* 文字列の順序を逆転 */
36:             save=string[i];
37:             string[i]=string[j];
38:             string[j]=save;
39:             j=j-1;
40:         }
41:     }
42:
43:     printf("文字列=%s¥n",string); /* 変換した文字列をプリント */
44: }
```

```
実行結果    整数=0  文字列=0
            整数=1  文字列=1
            整数=-1  文字列=-1
            整数=-123  文字列=-123
            整数=2345  文字列=2345
```

リスト2

```
 1: /*
 2:          整数→文字列(16進数)変換プログラム
 3: */
 4: int      number;                     /* ここに整数を読み込む       */
 5: int      sign;                       /* 符号を覚える                */
 6: char     string[100];                /* 変換結果をここに格納する */
 7:
 8: ma:n()
 9: {
10:     int  i,j;
11:     char save;
12:
13:     scanf("%d",&number);            /* 整数の読み込み */
14:     printf("整数=%d(%x) ",number,number); /* 読み込んだ値をプリント */
15:
16:     sign=0;
17:     if(number<0){                   /* 符号の記憶と絶対値 */
18:         sign=1;
19:         number=-number;
20:     }
21:     if(number==0)   strcpy(string,"0");
22:     else {
23:         for(i=0;number!=0;i=i+1){
24:             string[i]="0123456789ABCDEF"[number % 16];
25:                                     /* 16で割った余りを文字に */
26:             number=number / 16; /* 16で割った商    */
27:         }
28:         /* この時点で文字は逆順に並んでいる */
29:         if(sign==1){
30:             string[i]='-';          /* マイナス符号を追加 */
31:             i=i+1;
32:         }
33:         string[i]=0;                /* ヌル文字を入れて文字列に */
34:         j=i-1;                      /* 文字列の最後の文字の位置(ヌル文字以外) */
35:         for(i=0;i<j;i=i+1){       /* 文字列の順序を逆転 */
36:             save=string[i];
37:             string[i]=string[j];
38:             string[j]=save;
39:             j=j-1;
40:         }
41:     }
42:
43:     printf("文字列=%s¥n",string); /* 変換した文字列をプリント */
44: }
```

```
実行結果    整数=123456(1e240)  文字列=1E240
            整数=-123456(fffe1dc0)  文字列=-1E240
            整数=77777777(4a2cb71)  文字列=4A2CB71
```

だ '0' を加えるだけではだめなこと。

であることに注意すれば簡単にプログラムができてしまいます。それがリスト2です。

リスト2では数を16進数の文字に変換するために,

"0123456789ABCDEF"

というchar型配列(文字列)の要素を与えられた数を添字として取り出すという方法を使用しています(配列要素を取り出す [ ] という演算子は配列名だけでなく文字列定数につけてもいいのです)。このとき,添字0の要素は'0',添字1の要素は'1',……,添字15の要素は'F'ですからちょうど数を16進数に変換したことになっています。

*

今回はC言語のプログラムの基本となる文字列について説明しました。C言語では文字列の操作はすべてchar型配列の要素の操作として実現できるようになっています。配列の操作さえできれば文字列をどのように変形することも自由自在です。

C言語でアセンブラやコンパイラなどの基本プログラムを書く場合,まずファイルから1行分の文字列を入力してくるところから考えます。そしてその後は入力した文字列に操作を加えるだけです。規模の差こそあれアセンブラやコンパイラの作成と今回の例題のプログラムの本質に違いはありません(ちょっと嘘があるかな)。文字列の操作を修得することで皆さんのプログラミングの応用分野が大きく広がるのではないかと思います。

さて,次回は関数というものについて説明したいと思います。それでは来月までさようなら。

◆基礎力を高めようの解答

設問1

1) ○ 2) ○ 3) ○ 4) × 5) ○
6) ○ 7) ○ 8) ○ 9) ○ 10) ×

解説

2)3) 途中にヌル文字('¥0')があっても文字列である。ただし,ヌル文字以降の文字は存在しないものとみなされる。

4) 2重引用符を文字列に入れるためにはエスケープシーケンスを使う。この場合は"¥""と書く。

5) 文字列中の単一引用符ではエスケープシーケンスを使う必要はない。文字定数('¥'')として使うときにはエスケープシーケンスが必要。

6) この場合,2つの" "が並んだものと解釈される。文字列定数の連結規則により" "と同等である。ただし,XCのバージョン2ではエラーとみなされる。XCのバージョン1ではもともとコンパイルできない(ANSI Cでないため)。

7)8) 8進定数は0から7以外の文字が出てくると終了する。7)での8進定数の部分は¥01のみ,8)での8進定数の部分は¥012のみである。それ以外は通常の文字。

9) ¥c,¥d,¥eは未定義のエスケープシーケンス。これらは'c','d','e'という文字そのものと解釈される。

10) ¥xのあとには少なくとも1つの16進数が必要。

設問2

1) 0 2) 0 3) 3 5) 1
6) 0 7) 3 8) 2 9) 6

解説

エスケープシーケンスが1文字であることを考慮すれば難しくはないはず。

THE SENTINEL

●S-OSでもカードゲーム

このところ毎月のようにX68000用にはCARD DRVを使ったカードゲームが掲載されています。こういったものは内容としては共通システム向きのテーマといえます。これまでにも何度かカードゲーム用のシステムについて検討されたことはあるのですが，これまで実現されなかったのにはS-OSでトランプのマークが共通化できなかったことが大きく響いています。

今回のDOBONは各スートの頭文字を使い，また絵札はキャラクタグラフィックで構成して文字レベルでのカードゲームを実現しています。パターン認識は難しいのでカードの種類がわかりにくいかもしれませんが，慣れてくれば楽にカードの区別がつくようになります。

ゲームの内容も場に出せるカードの判定やドボンの判定など，徹底的に自動化されており，操作性は非常に良好です（ただ，プレイヤーのやることが少なすぎる気もしないではない)。コンピュータのリアクションもなかなか楽しいものがあります。

このゲームはSLANGで記述されています。ちょっとソースリストが大きいのですが，がんばって入力してみてください。

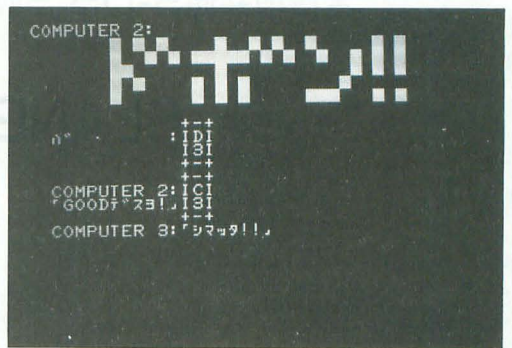
第105部
SLANG用カードゲームDOBON

●ついにC言語？

都合により一度は延期したS-OS“SWORD”へのC処理系の移植ですが，移植作業はほぼ完了しました。ちなみにSmall-C ver.2.7対応となっています。

さて，C言語を心待ちにしていた皆さん。WZDやWLKなどの準備はできていますか？　スクリーンエディタは揃っていますか？　以前予告したように，移植作業を行うためにはできるだけ簡単な方法を取っても，なおかなりの装備が必要になります。CP/M上のアセンブラも必要になりそうです。

テープでも十分な開発環境を実現するのがS-OSシステムの長所でもあったのですが，残念ながらもともとの処理系の性格上，ディスク専用となりカセットテープ，QDでは使用できません。

Small-Cは名前のとおりC言語としては小規模な処理系ですが，コンパイル後のオブジェクト，およびライブラリなどをあわせると結構大きなシステムとなります。完成すると2Dディスク4枚組になる予定です。発表は6/7月号くらいになりそうです。

●S-OSの系譜(20)

高速グラフィックパッケージMAGICをサポートした言語の登場によってS-OSでも簡単にグラフィックを扱えるようになりましたが，そもそもMAGICはS-OSの企画とは別に進行したものです。MZ-80Kなどハードウェア的にグラフィックを扱えないマシンも存在しており，S-OSの共通の世界ではやはりキャラクタ画面が統一された画面出力形態でした。そのなかで，キャラクタを使って疑似グラフィックを描くという試みが少なからずなされたのと並行して，文字を使ったゲームであるテキストアドベンチャーゲームが注目されたのは当然の動きといえるでしょう。

文章だけで作りあげられる冒険の世界はプレイするものの想像力をかき立て，独特の魅力を醸し出します。本誌でも清水和人氏の「テキストアドベンチャーを作ろう会」という企画がありましたし，名作として名高いCP/M-80のZORKシリーズが日本語で復活するという新しい動きもあります。

S-OSでは1987年2月号で，CONTEXというツールが発表されています。これは「テキストアドベンチャーを作ろう会」で掲載されたプログラムを下敷きに，完成したシナリオに従って，単語データ，表示するメッセージ，そしてそのコントロールプログラムを作成していくというアプローチで作られていました。このためツール上での試行錯誤は難しかったのですが，テキストアドベンチャーゲームを作るためのツールという新しい視点を見せてくれました。

1987年7月号ではこのCONTEXの流れを受け，STORY MASTERというプログラムが発表されました。CONTEXTは必要なデータを用意したあとコンバータで変換するいわばコンパイラのようなツールでしたが，STORY MASTERはコンピュータと対話しながらストーリーを作っていくインタプリタのようなツールです。テキストアドベンチャー用のtinyBASIC的なシステムで，エディタを内蔵しておりコツコツとゲームを広げていく，そんな作り方が可能となっていました。

全機種共通S-OS"SWORD"要

SLANG用カードゲームDOBON

グラフィックキャラクタはなくてもキャラクタグラフィックはできる。標準キャラクタのみを使ったカードゲームです。コンピュータのリアクションもなかなか楽しく、しかも全自動。ドボンの醍醐味を味わってください。

Ouchi Atsuo
大内 敦夫

S-OSとSLANGを使用したカードゲームを作りました。このDOBONというゲームは私が高校2年生のときクラスで(少し)流行したゲームです。そのとき以来,なんとかしてこれをコンピュータ上で走らせてみたいと思いたち,やっとこさ完成しました。カードの出し方はあの「UNO」にそっくりですがあがり方がまったく違い,それがこのゲームの面白さになっています。

編注) このゲームはトランプゲームのPAGE ONEをベースにした国産ゲームと思われますが,地方ルールが非常に多彩なのが特色です。今回のものは一般的と思われるものとも(おそらく)かなり違います。しかし,編集部内部でも統一見解が得られなかったため,原作者の送ってきたルールをそのまま採用しました。不満のある方は各自で改造してください。

ルールについて

カードはジョーカーを除いた52枚を使用します。プレイヤーは4人で4枚ずつの手札を配ります。残ったカードは山(以下パイルと呼ぶ)にして,いちばん上の1枚だけ表にして台札にします。

親から順番に台札と同じスートか同じ数字のカードを重ねていきます。出せなかったらパイルから手札に1枚加えます。

手持ちの札の合計が誰かの出したカードの数値と同じだったら「ドボン!」といって手札を場に開きます(ドボンでしかあがれません)。しかし,このときドボンした相手の手札の合計がいまドボンしたカードと同じだった場合,「ドボン返し!」といって逆転勝ちすることができます。

ドボン(またはドボン返し)した人は1枚手札を減らし,ドボンされた人は1枚手札を増やして次のゲームを始めます。ただし,都合上,ゲーム開始時の手札の数は1枚から11枚の範囲に制限されています。

●役札

以下のカードは特別な意味を持っています。

A:次の人の順番を飛ばす
2:次の人に2枚取らせる
3:次の人に3枚取らせる
8:スートを変える
9:逆回りにする

このうち,2と3は重ねていくことができます。前の人が2を出してきたとき,手札に同じスートの3があったとしましょう。ここで3を出すことができるのです。そして,さらに次の人は3を持っていない限り5枚のカードを引かなければなりません。もし3を持っていたなら,次の次の人は3か同じスートの2を持っていない限り8枚のカードを引かなければなりません。

プログラムについて

このプログラムはSLANGで記述してあります。ソースリストで31Kバイト,オブジェクトにしても23Kバイトという巨大なプログラムになってしまいました。

お手持ちのエディタからリストを入力してSLANGでコンパイルしてください。コンパイル直後のオブジェクトはオフセットがついていますのでそのままでは起動できません。3000_H番地に転送して起動してください。また,プログラムの最初とデモの最初にCONST宣言があり,時間待ちループの定数を設定しています。機種・好みによって変更してください。

ゲームが始まるとデモ画面となります。ルールの解説なども出ますのでわからない人は読んでおくといいでしょう。

画面表示は,いちばん下にユーザープレイヤーの手札,その上にメッセージ表示部,現在場に出ているカードのウィンドウ,そのほかにコンピュータプレイヤー4人分のウィンドウがあります。

次に操作法に関してですが,このゲームではどんな場合でも直接指定と選択指定の両方が可能です。

●直接指定

カードの選択時なら,A~R,その他の場合なら単語の頭文字(YESならY,HEARTならH)を押して指定する。

●選択指定

カーソルマークを4-6, Z-X, ←→のどれかで左右に動かし,スペースキー,リターンキーで指定する。カードの選択時には選択できるものだけにカーソルが移動します。このときリターンはパイルから1枚引くという意味になるので注意してください。

たいていの処理は自動的に判断して処理されます。

最後に

プログラム中のコメントにもあるとおり,このプログラムは最初FuzzyBASICで作っていました。しかし,速度面でお話にならず,コンパイラにかけたところ,なぜか

バグが出る始末。しかたがないので，Oh!MZ1986年11月号（あのMZ-700用tinyXCVIOUSが載っていた号です）にあったASCIIセーブプログラムを使ってASCIIファイルにし，それに手を加えて初期バージョンを完成させました。そういうこともあって，モジュール化が不徹底でプログラムが見にくいことが心残りなのですが……。

S-OSには疑似グラフィックキャラクタがほとんどありません。でも，私はドボンをコンピュータでやりたい一心でノーマルキャラクタだけでやり抜きました。特に絵札には（絵の下手な私としては）結構気をつかいました。なんとか顔らしく見えるもんでしょ？

それとコンピュータのレスポンスには多少ウケを狙ったものも入っています。これは多ければ多いほど楽しいと思うので，適当に追加するのも一興かと思います。あとはコンピュータの思考ルーチンがかなり手抜きなのでここを改造してもよいでしょう。

リスト1

```
  1 //
  2 // DOBON! BY A.OUCHI
  3 // TRANSLATED FROM Fuzzy BASIC TO SLANG
  4 //
  5 //*INITALIZE**********************
  6
  7   ORG     3000H;
  8   OFFSET  0A000H-3000H;
  9
 10 MACHINE LDIR(3);
 11
 12 CONST WLOOP=2500;
 13
 14 ARRAY !MSW[39],!CARD[51],!P_CARD[4][17],!P_LEFT[4],
 15       !P_START[4],!OWNER[51],!W_OR_L[4],!D_NO[4],
 16       !H_WL[4][4],!H_ST[4][4];
 17
 18 VAR PLAYER,DIR,OVER,PUT,REMAIN,
 19     CMARK,CNO,DCARD,WIN,LOSE,GAMES;
 20
 21 // *MAIN ROUTINE******************
 22 MAIN()
 23 BEGIN
 24   VAR I,J,DET;
 25   WIDTH(40);
 26   DEMO();
 27   FOR I=0 TO 4 (
 28     P_START[I]=4;
 29     FOR J=0 TO 4
 30       H_ST[I][J]=(H_WL[I][J]=0);
 31   )
 32   GAMES=0;
 33   PLAYER=RND(5);
 34   REPEAT (
 35     GAMEINIT();
 36     LOOP (
 37       PUT=0;
 38       DET=(PLAYER==0)?MANTURN():COMTURN();
 39       IF OVER==1 THEN EXIT;
 40       IF DET!=-1 THEN PUTCARD(PLAYER,DET);
 41       IF PLAYER==0 THEN (
 42         REORDER(0);
 43         DISPMYCARD();
 44       )
 45       IF DOBONCHECK() AND PUT THEN EXIT;
 46       DCARD=0;
 47       IF PUT==1 THEN SPECIAL();
 48       WAIT(5);
 49       COMMESCLEAR(PLAYER);
 50       IF DCARD!=0 THEN IF DRAW() THEN EXIT;
 51       NEXTPLAYER();
 52     )
 53     IF OVER==0 THEN DOBON();
 54   ) UNTIL GAMEOVER();
 55   PRINT(/,/,"GOOD-BYE!",/);
 56 END;
 57
 58 // *SUBROUTINES********************
 59
 60 // *INITIALIZE FOR EACH GAME*******
 61 GAMEINIT()
 62 BEGIN
 63   VAR I,N,JF;
 64   REPEAT (
 65     DIR=(OVER=(PUT=0));
 66     DISPINIT();
 67     FOR I=1 TO 4 (
 68       COMLOC(I,1,0);
 69       PRINT("COMPUTER ",I);
 70       COMMSG(I,5,1,"LEFT:0");
 71     )
 72     OWNER[0]=5;
 73     LDIR(OWNER,OWNER+1,51);
 74     SHUFFLE();
 75     FOR I=0 TO 4 (
 76       P_LEFT[I]=(W_OR_L[I]=0);
 77       DRAWCARD(I,P_START[I]);
 78     )
 79     REORDER(0);
 80     DISPMYCARD();
 81     I=REMAIN;
 82     REPEAT
 83       N=NUMBER(CARD[--I]);
 84     UNTIL (N!='A' AND N!='2' AND N!='3' AND N!='8' AND N!='9') OR I==-1;
 85     JF=0;
 86     IF I==-1 THEN (
 87       BEEP();
 88       MSGW("ｻﾞﾝﾈﾝﾅｶﾞﾗ､ｹﾞｰﾑｦ ｶｲｼﾃﾞｷﾏｾﾝ｡");
 89       MSGW("ｶｰﾄﾞｦ ｸﾊﾞﾘﾅｵｼﾏｽﾉﾃﾞ､ﾓｳｽｺｼ ｵﾏﾁｸﾀﾞｻｲ｡");
 90       JF=1;
 91     ) ELSE (
 92       N=CARD[I];
 93       IF I<--REMAIN THEN LDIR(CARD+I+1,CARD+I,REMAIN-I);
 94     )
 95   ) UNTIL JF==0;
 96   OWNER[N]=5;
 97   CMARK=MARK(N);
 98   CNO=NUMBER(N);
 99   DISPCARD();
100   LOCATE(21,2);
101   PRINT(FORM$(REMAIN,2));
102   SETMSG(0,"GAME START FROM ");
103   IF PLAYER==0
104     THEN SETMSG(16,"YOU");
105     ELSE (
106       SETMSG(16,"COMPUTER ");
107       MSW[25]='0'+PLAYER;
108       MSW[26]=0;
109     )
110   BEEP();
111   MSGW(MSW);
112 END;
113
114 // *DISPLAY INITIALIZE*************
115 DISPINIT()
116 BEGIN
117   VAR I;
118   PRINT("¥C",STR$('■',39),/);
119   FOR I=1 TO 15
120     IF I==8
121       THEN PRINT(STR$('■',15),SPC$(9),STR$('■',15),/);
122       ELSE PRINT("■",SPC$(13),"■         ■",SPC$(13),"■",/);
123   PRINT(STR$('■',39),/);
124   PRINT("■",SPC$(37),"■",/);
125   PRINT(STR$('■',39),/);
126   PRINT("■ A B C D E F G H I J K L M N O P Q R ■",/);
127   PRINT("■+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+■",/);
128   PRINT("■I I I I I I I I I I I I I I I I I I I■",/);
129   PRINT("■I I I I I I I I I I I I I I I I I I I■",/);
130   PRINT("■+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+■",/);
131   PRINT(STR$('■',39));
132   LOCATE(15,3);
133   PRINT("+-------+");
134   FOR I=4 TO 12 (
135     LOCATE(15,I);
136     PRINT("I       I");
137   )
138   LOCATE(15,13);
139   PRINT("+-------+");
140   LOCATE(16,2);
141   PRINT("PILE:52");
142   LOCATE(17,1);
143   PRINT("====>");
144   LOCATE(17,15);
145   PRINT("<====");
146 END;
147
148 // *PLAYER'S TURN*****************
149 MANTURN()
150 BEGIN
151   VAR DET,I;
152   MSG("ｱﾅﾀﾉ ﾊﾞﾝﾃﾞｽ｡");
153   BEEP();
154   WAIT(5);
155   IF P_LEFT[0]==1 THEN (
156     MSG("ｶｰﾄﾞｶﾞ 1ﾏｲｼｶ ｱﾘﾏｾﾝ｡");
157     WAIT(5);
158     MSG("PILEｶﾗ 1ﾏｲ ﾋｷﾏｽ｡");
159     WAIT(5);
160     MSGC();
161     DET=MANDRAW();
162   ) ELSE (
163     I=0;
164     WHILE CHECK(0,I)==FALSE AND I<P_LEFT[0]
165       I++;
166     IF I==P_LEFT[0] THEN (
167       MSG("ﾄﾞﾉ ｶｰﾄﾞﾓ ﾀﾞｾﾏｾﾝ｡");
168       WAIT(5);
169       MSG("PILEｶﾗ 1ﾏｲ ﾋｷﾏｽ｡");
170       WAIT(5);
171       MSGC();
172       DET=MANDRAW();
173     ) ELSE (
174       MSG("ﾄﾞﾚｦ ﾀﾞｼﾏｽｶ?  1ﾏｲ ﾋｸ=>RETURN");
175       DET=KEYIN(1);
176       IF DET==-1 THEN (
177         IF P_LEFT[0]==18
178           THEN TOOMANY();
179           ELSE DET=MANDRAW();
180       )
181     )
182   )
183 END(DET);
184
185 // *COMPUTER'S TURN***************
186 COMTURN()
187 BEGIN
188   VAR DET,I,DI,DF;
189   SETMSG(0,"COMPUTER ");
190   MSW[9]='0'+PLAYER;
191   SETMSG(10," ﾉ ﾊﾞﾝﾃﾞｽ｡");
192   MSG(MSW);
193   IF P_LEFT[PLAYER]==1 THEN (
194     COMMSG(PLAYER,1,2,"1ﾏｲｼｶ ﾓｯﾃ");
195     COMMSG(PLAYER,1,3,"ﾅｲｶﾗ､");
196     COMMSG(PLAYER,1,4,"PILEｶﾗ 1ﾏｲ");
197     COMMSG(PLAYER,1,5,"ﾋｸﾖ｡");
198     WAIT(5);
199     COMMESCLEAR(PLAYER);
200     DET=COMDRAW();
201   ) ELSE (
202     DET=-2;
203     COMLOC(PLAYER,1,3);
204     COMRES(3);
205     IF DIR==0 THEN I=(PLAYER==4)?0:PLAYER+1; ELSE I=PLAYER-1;
206     IF P_LEFT[I]+RND(5)<=2+P_LEFT[PLAYER]/2 THEN (
207       DF=9;
208       FOR I=0 TO P_LEFT[PLAYER]-1
209         IF CHECK(PLAYER,I) THEN
210           CASE NUMBER(P_CARD[PLAYER][I]) OF (
211             '3': DI=I,DF=1;
212             '2': IF DF>1 THEN DI=I,DF=2;
213             OTHERS: DF=3;
214           )
215       IF DF==9 THEN DET=COMDRAW();
216         ELSE IF DF<3 THEN DET=DI;
217     )
218     IF DET==-2 THEN (
219       DF=14;
220       FOR I=0 TO P_LEFT[PLAYER]-1
221         IF CHECK(PLAYER,I) THEN
222           CASE NUMBER(P_CARD[PLAYER][I]) OF (
223             'K': DI=I,DF=1;
224             'Q': IF DF>1 THEN DI=I,DF=2;
225             'J': IF DF>2 THEN DI=I,DF=3;
226             'T': IF DF>3 THEN DI=I,DF=4;
227             '7': IF DF>4 THEN DI=I,DF=5;
228             '9': IF DF>5 THEN DI=I,DF=6;
229             '6': IF DF>6 THEN DI=I,DF=7;
230             '5': IF DF>7 THEN DI=I,DF=8;
231             'A': IF DF>8 THEN DI=I,DF=9;
232             '4': IF DF>9 THEN DI=I,DF=10;
233             '8': IF DF>10 THEN DI=I,DF=11;
234             '2': IF DF>11 THEN DI=I,DF=12;
235             '3': IF DF>12 THEN DI=I,DF=13;
236           )
```

```
237       IF DF==14
238         THEN DET=COMDRAW();
239         ELSE (
240           DET=DI;
241           WAIT(2);
242           COMMESCLEAR(PLAYER);
243         )
244     )
245   )
246   WAIT(1);
247 END(DET);
248
249 // *SPECIAL CARDS******************
250 SPECIAL()
251 BEGIN
252   VAR I,D,P,SP,HT,DM,CL;
253   CASE CNO OF (
254     '9': (
255           MSG("REVERSE!");
256           DIR=1-DIR;
257           LOCATE(17,1);
258           IF DIR THEN PRINT("<===="); ELSE PRINT("====>");
259           LOCATE(17,15);
260           IF DIR THEN PRINT("====>"); ELSE PRINT("<====");
261         )
262     'A': (
263           MSG("SKIP!");
264           COMMESCLEAR(PLAYER);
265           NEXTPLAYER();
266         )
267     '8': (
268           IF PLAYER==0 THEN (
269             MSG("ｽｰﾄﾊ ﾅﾆ?  SPADE/HEART/DIAMOND/CLUB");
270             D=FALSE;
271             P=0;
272             REPEAT (
273               LOCATE(P*6+12+(P==3)*2,17);
274               CASE INKEY(1) OF (
275                 'S': D=TRUE,P=0;
276                 'H': D=TRUE,P=1;
277                 'D': D=TRUE,P=2;
278                 'C': D=TRUE,P=3;
279                 '4','¥L','Z': P=(P==0)?3:P-1;
280                 '6','¥R','X': P=(P==3)?0:P+1;
281                 ' ','¥N': D=TRUE;
282               )
283             ) UNTIL D;
284             CMARK=MARK(P*13);
285             BEEP();
286           ) ELSE (
287             IF P_LEFT[PLAYER]>1 THEN (
288               SP=(HT=(DM=(CL=0)));
289               FOR I=0 TO P_LEFT[PLAYER]-1
290                 CASE P_CARD[PLAYER][I]/13 OF (
291                   0: SP++;
292                   1: HT++;
293                   2: DM++;
294                   3: CL++;
295                 )
296               CMARK='S';
297               IF SP<HT THEN SP=HT,CMARK='H';
298               IF SP<DM THEN SP=DM,CMARK='D';
299               IF SP<CL THEN CMARK='C';
300             ) ELSE (
301               REPEAT
302                 CMARK=MARK(RND(4)*13);
303               UNTIL CMARK!=MARK(P_CARD[PLAYER][0]);
304             )
305             COMMESCLEAR(PLAYER);
306             COMMSG(PLAYER,0,2,"ｽｰﾄﾊ ");
307             CASE CMARK OF (
308               'S': PRINT("SPADE.");
309               'H': PRINT("HEART.");
310               'D': PRINT("DIAMOND.");
311               'C': PRINT("CLUB.");
312             )
313           )
314           MSG("ALMIGHTY!");
315           LOCATE(18,14);
316           PRINT("<",CHR$(CMARK),">");
317         )
318     '2': DCARD=2;
319     '3': DCARD=3;
320   )
321 END;
322
323 // *DRAW 2/3 CARDS*****************
324 DRAW()
325 BEGIN
326   VAR FF,I,NM,DF,DI;
327   REPEAT (
328     IF PLAYER==0 THEN (
329       SETMSG(0,"DRAW ");
330       IF DCARD>9 THEN (
331         MSW[5]='0'+DCARD/10;
332         MSW[6]='0'+DCARD MOD 10;
333         MSW[7]='!';
334         MSW[8]=0;
335       ) ELSE (
336         MSW[5]='0'+DCARD;
337         MSW[6]='!';
338         MSW[7]=0;
339       )
340       MSG(MSW);
341       BEEP();
342       WAIT(5);
343       MSGC();
344     ) ELSE (
345       COMLOC(PLAYER,1,3);
346       PRINT("DRAW ",DCARD,"!");
347       WAIT(5);
348       COMMSG(PLAYER,1,3,"         ");
349     )
350     NEXTPLAYER();
351     IF PLAYER==0 THEN (
352       MSG("ｱﾅﾀﾉ ﾊﾞﾝﾃﾞｽ｡");
353       BEEP();
354       WAIT(5);
355     ) ELSE (
356       SETMSG(0,"COMPUTER ");
357       MSW[9]='0'+PLAYER;
358       SETMSG(10," ﾉ ﾊﾞﾝﾃﾞｽ｡");
359       MSG(MSW);
360     )
361     FF=(I=0);
362     REPEAT (
363       NM=NUMBER(P_CARD[PLAYER][I]);
364       IF CHECK(PLAYER,I) AND (NM=='2' OR NM=='3') THEN FF=1;
365     ) UNTIL ++I==P_LEFT[PLAYER] OR FF==1;
366     IF FF==0 OR P_LEFT[PLAYER]==1 THEN (
367       FF=0;
368       IF PLAYER==0 THEN (
369         MSG("ﾄﾞﾉ ｶｰﾄﾞﾓ ﾀﾞｾﾏｾﾝ｡");
370         WAIT(5);
371       )
372       DRAWMANY(PLAYER,DCARD);
373       IF PLAYER==0 THEN (
374         REORDER(0);
375         DISPMYCARD();
376       )
377     ) ELSE (
378       IF PLAYER==0 THEN (
379         MSG("ﾄﾞﾚｦ ﾀﾞｼﾏｽｶ?  ﾋｸ=>RETURN");
380         REPEAT (
381           I=KEYIN(0);
382           IF I!=-1 THEN NM=NUMBER(P_CARD[0][I]);
383         ) UNTIL NM=='2' OR NM=='3' OR I==-1;
384         IF I==-1 THEN (
385           DRAWMANY(0,DCARD);
386           FF=0;
387         ) ELSE (
388           PUTCARD(0,I);
389           DCARD=DCARD+NM-'0';
390         )
391         REORDER(0);
392         DISPMYCARD();
393       ) ELSE (
394         FOR I=0 TO P_LEFT[PLAYER]-1 (
395           DF=0;
396           IF CHECK(PLAYER,I) THEN
397             CASE NUMBER(P_CARD[PLAYER][I]) OF (
398               '3': DI=I,DF=1;
399               '2': IF DF==0 THEN DI=I,DF=2;
400             )
401         )
402         DCARD=DCARD+NUMBER(P_CARD[PLAYER][DI])-'0';
403         WAIT(3);
404         PUTCARD(PLAYER,DI);
405       )
406     )
407     IF OVER==1 THEN EXIT;
408     IF PLAYER!=0 THEN COMMESCLEAR(PLAYER);
409     IF FF==1 AND DOBONCHECK() THEN EXIT;
410   ) UNTIL FF==0;
411 END(FF==1 OR OVER==1);
412
413 // *NEXT PLAYER*********************
414 NEXTPLAYER()
415 BEGIN
416   IF DIR==0
417     THEN PLAYER=(PLAYER==4)?0:PLAYER+1;
418     ELSE PLAYER=PLAYER?PLAYER-1:4;
419 END;
420
421 // *DOBON!!************************
422 DOBON()
423 BEGIN
424   VAR I,DB,WP;
425   ARRAY !WIN[3];
426   PRINT("¥C");
427   WP=0;
428   FOR I=0 TO 4
429     IF W_OR_L[I]==1 THEN (
430       DB=D_NO[I];
431       WIN[WP++]=I;
432       IF I==0
433         THEN PRINT("YOU,");
434         ELSE PRINT("COMPUTER ",I,",");
435     )
436   PRINT("¥L:",/);
437   PRINT(TAB$(7),"■■■  ■ ■■ ■   ■■",/);
438   PRINT(TAB$(7),"■ ■■■■■■■■ ■ ■■■",/);
439   PRINT(TAB$(7),"■■    ■       ■■■",/);
440   PRINT(TAB$(7),"■■   ■■■    ■    ",/);
441   PRINT(TAB$(7),"■    ■■■   ■■  ■■",/,/);
442   PRINT("                +-+",/);
443   PRINT("  ﾊﾞ            :I",CHR$(CMARK),"I",/);
444   PRINT("                I",CHR$(CNO),"I",/);
445   PRINT("                +-+");
446   FOR I=0 TO WP-1 CARDOPEN(I,WIN[I]);
447   PRINT(/,/);
448   BEEP(); BEEP();
449   WAIT(5);
450   IF D_NO[PLAYER]==DB THEN (
451     PRINT("¥C");
452     IF PLAYER==0
453       THEN PRINT("YOU:",/);
454       ELSE PRINT("COMPUTER ",PLAYER,":",/);
455     PRINT("     ■■■  ■ ■■ ■   ",/);
456     PRINT("     ■ ■■■■■■■ ■ ■",/);
457     PRINT("     ■■    ■       ■",/);
458     PRINT("     ■■   ■■■     ■ ",/);
459     PRINT("     ■    ■■■   ■■  ",/,/);
460     PRINT(TAB$(10)," ■ ■■ ■■■■■■  ■■",/);
461     PRINT(TAB$(10),"■■■■■ ■■  ■   ■■■",/);
462     PRINT(TAB$(10)," ■■     ■ ■■ ■■■",/);
463     PRINT(TAB$(10)," ■ ■    ■   ■    ",/);
464     PRINT(TAB$(10),"■ ■■   ■■■■■■  ■■");
465     CARDOPEN(1,PLAYER);
466     BEEP(); BEEP(); BEEP();
467     IF PLAYER!=0 THEN (
468       LOCATE(2,16);
469       COMRES(1);
470     )
471     FOR I=0 TO WP-1 (
472       LOCATE(2,I+19-(WIN[0]==0));
473       IF WIN[I]!=0 THEN (
474         PRINT("COMPUTER ",WIN[I],":｢");
475         COMRES(2);
476         PRINT("｣");
477       )
478       W_OR_L[WIN[I]]=2;
479     )
480     W_OR_L[PLAYER]=1;
481   ) ELSE (
482     FOR I=0 TO WP-1
483       IF WIN[I]!=0 THEN (
484       LOCATE(2,I*3+13);
485       COMRES(1);
486       )
487     W_OR_L[PLAYER]=2;
488     IF PLAYER!=0 THEN (
489       LOCATE(2,WP*3+12);
490       PRINT("COMPUTER ",PLAYER,":｢");
491       COMRES(2);
492       PRINT("｣");
493     )
494   )
495 END;
496
497 // *GAME OVER**********************
498 GAMEOVER()
499 BEGIN
500   VAR I,J,R,SC;
501   WAIT(12);
502   PRINT("¥C        ****** GAME OVER ******¥N¥N");
503   IF GAMES>4 THEN
504     FOR I=0 TO 4
505       FOR J=0 TO 4 (
506         H_WL[I][J]=H_WL[I][J+1];
507         H_ST[I][J]=H_ST[I][J+1];
508       )
509   R=(GAMES>4)?4:GAMES;
510   WIN=(LOSE=0);
511   FOR I=0 TO 4
512     CASE H_WL[I][R]=W_OR_L[I] OF (
513       1: WIN++;
514       2: LOSE++;
515     )
516   FOR I=0 TO 4 (
517     CASE W_OR_L[I] OF (
518       1: (
519             P_START[I]=P_START[I]-LOSE;
520             IF P_START[I].<=.0 THEN P_START[I]=1;
521           )
522       2: (
523             P_START[I]=P_START[I]+WIN+(WIN==0);
524             IF P_START[I]>10 THEN P_START[I]=10;
```


▶やっと退院した！　それにしても49日間の絶食はきつかった。皆さん，胃腸は大切にしてくださいね。
鈴木　政宏(18)宮城県

```
525           )
526     )
527     H_ST[I][R]=P_START[I];
528   )
529   R=(GAMES<5)?0:GAMES-4;
530   PRINT(" +-----+-----+-----+-----+-----+-----+¥N");
531   PRINT(" I      I");
532   FOR I=1 TO 5
533     PRINT(" ",FORM$(R+I,3)," I");
534   PRINT("¥N +-----+-----+-----+-----+-----+-----+¥N");
535   FOR I=0 TO 4 (
536     PRINT(" I");
537     IF I==0 THEN PRINT(" YOU "); ELSE PRINT("COM ",I);
538     PRINT("I");
539     FOR J=0 TO 4 (
540       IF R+J<=GAMES THEN (
541         CASE H_WL[I][J] OF (
542           0: PRINT("   ");
543           1: PRINT("^.^");
544           2: PRINT(";.;");
545         )
546         PRINT(FORM$(H_ST[I][J],2));
547       ) ELSE PRINT(" --- ");
548       PRINT("I");
549     ) PRINT("¥N +-----+-----+-----+-----+-----+-----+¥N");
550   )
551   PRINT("          ^.^ = WIN    ;.; = LOSE¥N");
552   PRINT(/,"    DO YOU WANT TO TRY AGAIN?");
553   R=YESNO(29,17,TRUE);
554   IF R THEN (
555     CASE WIN OF (
556       0: (
557             REPEAT
558               PLAYER=RND(5);
559             UNTIL W_OR_L[PLAYER]==0;
560           )
561       1: FOR I=0 TO 4 IF W_OR_L[I]==1 THEN PLAYER=I;
562       OTHERS: (
563                 I=RND(WIN)+1;
564                 SC=0;
565                 FOR J=0 TO 4
566                   IF W_OR_L[J]==1 THEN IF ++SC==I THEN PLAYER=J;
567               )
568     )
569     GAMES++;
570     PRINT(/,/,"   SCORE RESET?  ");
571     IF YESNO(18,19,FALSE) THEN (
572       GAMES=0;
573       FOR I=0 TO 4 (
574         P_START[I]=4;
575         FOR J=0 TO 4
576           H_ST[I][J]=(H_WL[I][J]=0);
577       )
578     )
579   )
580 END(1-R);
581
582 // *DRAW 1 CARD*******************
583 DRAW1(PL)
584 BEGIN
585   VAR J;
586   J=CARD[--REMAIN];
587   OWNER[J]=PL;
588   P_CARD[PL][P_LEFT[PL]++]=J;
589   IF REMAIN==0 THEN SHUFFLE();
590   LOCATE(21,2);
591   PRINT(FORM$(REMAIN,2));
592   IF PL==0 THEN (
593     LOCATE(P_LEFT[0]*2,21);
594     PRINT(CHR$(MARK(P_CARD[0][P_LEFT[0]-1])),"¥D¥L",
595           CHR$(NUMBER(P_CARD[0][P_LEFT[0]-1])));
596   ) ELSE (
597     COMLOC(PL,10,1);
598     PRINT(P_LEFT[PL]);
599   )
600 END;
601
602 // *DRAW CARDS*********************
603 DRAWCARD(PL,NO)
604 BEGIN
605   VAR I;
606   FOR I=1 TO NO DRAW1(PL);
607 END;
608
609 // *DRAW MANY CARDS****************
610 DRAWMANY(PL,NO)
611 BEGIN
612   IF PL==0 THEN (
613     IF NO>9 THEN (
614       MSW[0]=NO/10+'0';
615       MSW[1]=NO MOD 10+'0';
616       SETMSG(2,"ﾏｲ ﾋｷﾏｽ｡");
617     ) ELSE (
618       MSW[0]=NO+'0';
619       SETMSG(1,"ﾏｲ ﾋｷﾏｽ｡");
620     )
621     MSG(MSW);
622     WAIT(5);
623   ) ELSE (
624     COMLOC(PL,1,2); COMRES(2);
625     COMLOC(PL,1,3); PRINT(NO,"ﾏｲ");
626     COMMSG(PL,1,4,"ﾋｶﾅｷｬ....");
627   )
628   IF P_LEFT[PL]+NO<=18 THEN (
629     DRAWCARD(PL,NO);
630     WAIT(5);
631     IF PL!=0 THEN COMMESCLEAR(PL);
632   ) ELSE (
633     DRAWCARD(PL,18-P_LEFT[PL]);
634     IF PL==0 THEN TOOMANY(); ELSE GIVEUP();
635   )
636 END;
637
638 // *CARD SHUFFLE******************
639 SHUFFLE()
640 BEGIN
641   VAR I,J,K,TEMP;
642   REMAIN=0;
643   FOR I=0 TO 51
644     IF OWNER[I]==5 THEN OWNER[CARD[REMAIN++]=I]=-1;
645   IF REMAIN==0 THEN (
646     BEEP();
647     MSGW("ｶｰﾄﾞｶﾞ ﾅｸﾅｯﾃ ｼﾏｲﾏｼﾀ｡");
648     MSGW("ｻﾞﾝﾈﾝﾃﾞｽｶﾞ､ｹﾞｰﾑｦ ﾁｭｳｼｼﾏｽ｡");
649     STOP();
650   )
651   FOR I=0 TO REMAIN*5 (
652     TEMP=CARD[J=RND(REMAIN)];
653     CARD[J]=CARD[K=RND(REMAIN)];
654     CARD[K]=TEMP;
655   )
656 END;
657
658 // *MAN DRAWS 1 CARD**************
659 MANDRAW()
660 BEGIN
661   VAR DET,K;
662   DRAW1(0);
663   DISPMYCARD();
664   DET=-1;
665   IF CHECK(0,P_LEFT[0]-1) THEN (
666     MSW[0]='¥"';
667     MSW[1]='@'+P_LEFT[0];
668     SETMSG(2,"¥" ｦ ﾀﾞｼﾏｽｶ?  ");
669     MSG(MSW);
670     IF YESNO(17,17,TRUE) THEN DET=P_LEFT[0]-1;
671   ) ELSE WAIT(2);
672 END(DET);
673
674 // *COMPUTER DRAWS 1 CARD*********
675 COMDRAW()
676 BEGIN
677   VAR DET;
678   COMMESCLEAR(PLAYER);
679   IF P_LEFT[PLAYER]==18
680     THEN GIVEUP();
681     ELSE (
682       COMMSG(PLAYER,1,2,"1ﾏｲ ﾋｸﾖ｡");
683       DRAW1(PLAYER);
684       IF CHECK(PLAYER,P_LEFT[PLAYER]-1) THEN (
685         COMLOC(PLAYER,1,4);
686         COMRES(4);
687         PRINT("ﾀﾞｾﾙｾﾞ｡");
688         DET=P_LEFT[PLAYER]-1;
689       ) ELSE (
690         COMLOC(PLAYER,2,4);
691         COMRES(5);
692         PRINT("ﾊﾟｽ...");
693         DET=-1;
694       )
695     )
696 END(DET);
697
698 // *TOO MANY CARDS!!**************
699 TOOMANY()
700 BEGIN
701   BEEP();
702   MSGW("ｺﾚｲｼﾞｮｳ ｶｰﾄﾞｦ ﾓﾃﾏｾﾝ!!");
703   MSGW("ｱﾅﾀﾉ ﾏｹﾃﾞｽ!!");
704   W_OR_L[0]=2;
705   OVER=1;
706 END;
707
708 // *COMPUTER GIVE UP**************
709 GIVEUP()
710 BEGIN
711   COMMESCLEAR(PLAYER);
712   BEEP();
713   COMLOC(PLAYER,0,2); COMRES(2);
714   COMMSG(PLAYER,0,3,"ｺﾚｲｼﾞｮｳ");
715   COMMSG(PLAYER,0,4,"ｶｰﾄﾞｦ");
716   COMMSG(PLAYER,0,5,"ﾓﾃﾅｲﾖ!!");
717   COMMSG(PLAYER,0,6,"GIVE UP...");
718   W_OR_L[PLAYER]=2;
719   OVER=1;
720 END;
721
722 // *PUT A CARD********************
723 PUTCARD(PL,NO)
724 BEGIN
725   VAR CD;
726   LOCATE(18,14);
727   PRINT("   ");
728   CD=P_CARD[PL][NO];
729   CMARK=MARK(CD);
730   CNO=NUMBER(CD);
731   DISPCARD();
732   PUT=1;
733   OWNER[CD]=5;
734   IF --P_LEFT[PL]>NO
735     THEN LDIR(P_CARD+PL*18+NO+1,P_CARD+PL*18+NO,P_LEFT[PL]-NO);
736   IF PL!=0 THEN (
737     COMLOC(PL,10,1);
738     PRINT(P_LEFT[PL]," ");
739   )
740   BEEP();
741 END;
742
743 // *DISPLAY CARD******************
744 DISPCARD()
745 BEGIN
746   VAR I;
747   LOCATE(16,4);
748   PRINT(CHR$(CNO));
749   LOCATE(22,12);
750   PRINT(CHR$(CNO));
751   PUTMK(16,5);
752   PUTMK(22,11);
753   FOR I=5 TO 11 (
754     LOCATE(17,I);
755     PRINT("      ");
756   )
757   CASE CNO OF (
758     'A': PUTMK(19,8);
759     '2': PIC2();
760     '3': PIC3();
761     '4': PIC4();
762     '5': PIC5();
763     '6': PIC6();
764     '7': PIC7();
765     '8': PIC8();
766     '9': PIC9();
767     'T': PIC10();
768     'J': PICJACK();
769     'Q': PICQUEEN();
770     'K': PICKING();
771   )
772 END;
773
774 PIC2()
775 BEGIN
776   PUTMK(19,6);
777   PUTMK(19,10);
778 END;
779
780 PIC3()
781 BEGIN
782   PUTMK(19,8);
783   PIC2();
784 END;
785
786 PIC4()
787 BEGIN
788   PUTMK(18,6);
789   PUTMK(20,6);
790   PUTMK(18,10);
791   PUTMK(20,10);
792 END;
793
794 PIC5()
795 BEGIN
796   PUTMK(19,8);
797   PIC4();
798 END;
799
800 PIC6()
801 BEGIN
802   PUTMK(18,8);
803   PUTMK(20,8);
804   PIC4();
805 END;
806
807 PIC7()
808 BEGIN
809   PUTMK(17,5);
810   PUTMK(21,5);
811   PUTMK(17,7);
812   PUTMK(21,7);
```


▶とうとう卒業だ。いま思えば本当に早かった。このような気持ちは誰にでもあるのだろうなあ。

谷本　善孝(18)北海道

```
813   PUTMK(17,11);
814   PUTMK(21,11);
815   PUTMK(19,6);
816 END;
817
818 PIC8()
819 BEGIN
820   PUTMK(17,5);
821   PUTMK(21,5);
822   PUTMK(17,8);
823   PUTMK(21,8);
824   PUTMK(17,11);
825   PUTMK(21,11);
826   PUTMK(19,6);
827   PUTMK(19,10);
828 END;
829
830 PIC910SUB()
831 BEGIN
832   PUTMK(17,5);
833   PUTMK(21,5);
834   PUTMK(17,7);
835   PUTMK(21,7);
836   PUTMK(17,9);
837   PUTMK(21,9);
838   PUTMK(17,11);
839   PUTMK(21,11);
840 END;
841
842 PIC9()
843 BEGIN
844   PUTMK(19,8);
845   PIC910SUB();
846 END;
847
848 PIC10()
849 BEGIN
850   PUTMK(19,6);
851   PUTMK(19,10);
852   PIC910SUB();
853 END;
854
855 PICJACK()
856 BEGIN
857   LOCATE(17,5);   PRINT(" - - ");
858   LOCATE(17,6);   PRINT("     ");
859   LOCATE(17,7);   PRINT(" ",CHR$(CMARK)," ",CHR$(CMARK)," ");
860   LOCATE(17,8);   PRINT("     ");
861   LOCATE(17,9);   PRINT("  J  ");
862   LOCATE(17,10);  PRINT("     ");
863   LOCATE(17,11);  PRINT(" --- ");
864 END;
865
866 PICQUEEN()
867 BEGIN
868   LOCATE(17,5);   PRINT(" ^ ^ ");
869   LOCATE(17,6);   PRINT("     ");
870   LOCATE(17,7);   PRINT(" Q Q ");
871   LOCATE(17,8);   PRINT("     ");
872   LOCATE(17,9);   PRINT("( ",CHR$(CMARK)," )");
873   LOCATE(17,10);  PRINT("     ");
874   LOCATE(17,11);  PRINT(" --/ ");
875 END;
876
877 PICKING()
878 BEGIN
879   LOCATE(17,5);   PRINT(" V V ");
880   LOCATE(17,6);   PRINT("     ");
881   LOCATE(17,7);   PRINT(" ",CHR$(CMARK)," ",CHR$(CMARK)," ");
882   LOCATE(17,8);   PRINT("     ");
883   LOCATE(17,9);   PRINT("/ K /");
884   LOCATE(17,10);  PRINT(".....");
885   LOCATE(17,11);  PRINT(" >-< ");
886 END;
887
888 // *PUT A MARK**********************
889 PUTMK(X,Y)
890 BEGIN
891   LOCATE(X,Y);
892   PRINT(CHR$(CMARK));
893 END;
894
895 // *MARK****************************
896 MARK(CD)
897 BEGIN
898   VAR MK;
899   CASE CD/13 OF (
900     0: MK='S';
901     1: MK='H';
902     2: MK='D';
903     3: MK='C';
904   )
905 END(MK);
906
907 // *NUMBER**************************
908 NUMBER(CD)
909 BEGIN
910   VAR NUM;
911   CASE CD MOD 13 OF (
912     0     : NUM='A';
913     1 TO 8: NUM='1'+CD MOD 13;
914     9     : NUM='T';
915     10    : NUM='J';
916     11    : NUM='Q';
917     12    : NUM='K';
918   )
919 END(NUM);
920
921 // *DISPLAY MESSAGE AND PAUSE******
922 MSGW(ADR)
923 BEGIN
924   VAR DUMMY;
925   MSG(ADR);
926   DUMMY=INKEY(1);
927   MSGC();
928 END;
929
930 // *MESSAGE CLEAR******************
931 MSGC()
932 BEGIN
933   LOCATE(1,17);
934   PRINT(SPC$(37));
935 END;
936
937 // *DISPLAY MESSAGE WITHOUT PAUSE**
938 MSG(ADR)
939 BEGIN
940   MSGC();
941   LOCATE(2,17);
942   PRINT(MSX$(ADR));
943 END;
944
945 // *SET MESSAGE TO MSW[]***********
946 SETMSG(PO,ADR)
947 BEGIN
948   WHILE MSW[PO++]=MEM[ADR++] ;
949   MSW[PO]=0;
950 END;
951
952 // *COMPUTERS' MESSAGE*************
953 COMMSG(PL,DX,DY,ADR)
954 BEGIN
955   COMLOC(PL,DX,DY);
956   PRINT(MSX$(ADR));
957 END;
958
959 // *LOCATE FOR COMPUTER************
960 COMLOC(COMNO,DX,DY)
961 BEGIN
962   LOCATE((COMNO<=2?1:25)+DX,((COMNO==2 OR COMNO==3)?1:9)+DY);
963 END;
964
965 // *COMPUTER MESSAGE CLEAR*********
966 COMMESCLEAR(PL)
967 BEGIN
968   VAR I;
969   FOR I=2 TO 6
970     COMMSG(PL,0,I,"              ");
971 END;
972
973 // *COMPUTERS' RESPONSE************
974 COMRES(PAT)
975 BEGIN
976   CASE PAT OF (
977 // *WIN!****************************
978     1: CASE RND(7) OF (
979          0: PRINT("｢ﾔｯﾀｾﾞ!!｣");
980          1: PRINT("｢ﾗｯｷｰ!｣");
981          2: PRINT("｢ﾜｯﾊｯﾊｯ･･･｣");
982          3: PRINT("｢ｻﾞﾏｰﾐﾛ!｣");
983          4: PRINT("｢ｶｯﾀｰ!!｣");
984          5: PRINT("｢ﾜﾀｼﾊ ﾀﾞﾚﾉ¥D¥L¥L¥L¥L¥L¥L¥L¥Lﾁｮｳｾﾝﾃﾞﾓ ｳｹﾙ!!｣");
985          6: PRINT("｢GOODﾃﾞｽﾖ!｣");
986        )
987 // *LOST/DRAW/GIVE UP**************
988     2: CASE RND(8) OF (
989          0: PRINT("ｹﾞｹﾞｯ!!");
990          1: PRINT("ｼﾏｯﾀ!!");
991          2: PRINT("ｱﾘｬ!?");
992          3: PRINT("ﾎﾞｯｸﾝ｡");
993          4: PRINT("ﾔﾗﾚﾀ!!");
994          5: PRINT("OH,NO!!");
995          6: PRINT("ｹﾞﾛｹﾞﾛ･･･");
996          7: PRINT("ﾚﾚﾚﾚﾚﾚ･･･");
997        )
998 // *THINKING************************
999     3: CASE RND(6) OF (
1000         0: PRINT("ｴｰﾄ･･･");
1001         1: PRINT("ｳｰﾑ･･･");
1002         2: PRINT("ﾄﾞｳｼﾖｳ･･･");
1003         3: PRINT("ﾏｲｯﾀﾅ･･･");
1004         4: PRINT("ｿｳﾀﾞﾅ･･･");
1005         5: PRINT("ﾑﾑﾑ･･･");
1006       )
1007 // *LUCKY!**************************
1008    4: CASE RND(3) OF (
1009         0: PRINT("OK!");
1010         1: PRINT("ﾖｼ!");
1011         2: PRINT("ﾌﾌｯ｡");
1012       )
1013 // *UNLUCKY...**********************
1014    5: CASE RND(5) OF (
1015         0: PRINT("ｻﾞﾝﾈﾝ!¥D¥L¥L¥L");
1016         1: PRINT("ﾁｯ!");
1017         2: PRINT("ｹｯ!");
1018         3: PRINT("ｱﾚ?");
1019         4: PRINT("...");
1020       )
1021   )
1022 END;
1023
1024 // *DISPLAY PLAYER'S CARDS*********
1025 DISPMYCARD()
1026 BEGIN
1027   VAR I;
1028   FOR I=1 TO 18 (
1029     LOCATE(I*2,21);
1030     IF I>P_LEFT[0]
1031       THEN PRINT(" ¥D¥L ");
1032       ELSE PRINT(CHR$(MARK(P_CARD[0][I-1])),"¥D¥L",
1033                  CHR$(NUMBER(P_CARD[0][I-1])));
1034   )
1035 END;
1036
1037 // *REORDER CARDS (INSERTION SORT)*
1038 REORDER(PL)
1039 BEGIN
1040   VAR J,CD,P;
1041   IF P_LEFT[PL]==1 THEN RETURN;
1042   FOR J=1 TO P_LEFT[PL]-1 (
1043     CD=P_CARD[PL][J];
1044     P=J;
1045     WHILE CD<P_CARD[PL][P-1] AND P>0
1046       P_CARD[PL][P]=P_CARD[PL][--P];
1047     P_CARD[PL][P]=CD;
1048   )
1049 END;
1050
1051 // *COMPUTER'S CARD OPEN***********
1052 CARDOPEN(POS,PL)
1053 BEGIN
1054   VAR I;
1055   REORDER(PL);
1056   LOCATE(2,POS*3+12);
1057   IF PL==0
1058     THEN PRINT("YOU        :");
1059     ELSE PRINT("COMPUTER ",PL,":");
1060   LOCATE(13,POS*3+11);
1061   PRINT("+¥D¥LI¥D¥LI¥D¥L+");
1062   FOR I=0 TO P_LEFT[PL]-1 (
1063     LOCATE(I*2+14,POS*3+11);
1064     PRINT("-+¥D¥L¥L",CHR$(MARK(P_CARD[PL][I])),"I¥D¥L¥L",
1065           CHR$(NUMBER(P_CARD[PL][I])),"I¥D¥L¥L-+");
1066   )
1067 END;
1068
1069 // *CHECK IF THE CARD CAN PUT******
1070 CHECK(PL,NO)
1071 BEGIN
1072   VAR CD;
1073   CD=P_CARD[PL][NO];
1074 END(MARK(CD)==CMARK OR NUMBER(CD)==CNO);
1075
1076 // *CHECK WHETHER "DOBON" OR NOT***
1077 DOBONCHECK()
1078 BEGIN
1079   VAR I,J,DF;
1080   DF=0;
1081   FOR I=0 TO 4 (
1082     W_OR_L[I]=0;
1083     D_NO[I]=0;
1084     FOR J=0 TO P_LEFT[I]-1
1085       D_NO[I]=D_NO[I]+P_CARD[I][J] MOD 13+1;
1086     IF I!=PLAYER AND D_NO[I]<=13 AND NUMBER(D_NO[I]-1)==CNO
1087       THEN W_OR_L[I]=(DF=1);
1088   )
1089 END(DF);
1090
1091 // *KEY IN**************************
1092 KEYIN(M)
1093 BEGIN
1094   VAR I,P,K,N;
1095   P=-1;
1096   REPEAT (
1097     P=(P+1)*(P!=P_LEFT[0]-1);
1098     N=NUMBER(P_CARD[0][P]);
1099   ) UNTIL CHECK(0,P) AND (M OR N=='2' OR N=='3');
1100   I=FALSE;
```

▶ちゃだワもすっかり定着して，毎年のイベントとして楽しませてもらっています。なんか，ちゃだワが来るのが１年の始まりみたいでいいですね。　坂本　秀司(19)宮城県

```
1101   REPEAT (
1102     LOCATE(P*2+2,19);
1103     K=INKEY(1);
1104     CASE K OF (
1105          'A' TO 'R': IF CHECK(0,K-'A') THEN P=K-'A',I=TRUE;
1106        '4','¥L','Z': (
1107                          REPEAT (
1108                            P=P-1+(P==0)*P_LEFT[0];
1109                            N=NUMBER(P_CARD[0][P]);
1110                          ) UNTIL CHECK(0,P) AND (M OR N=='2' OR N=='3');
1111                        )
1112        '6','¥R','X': (
1113                          REPEAT (
1114                            P=(P+1)*(P!=P_LEFT[0]-1);
1115                            N=NUMBER(P_CARD[0][P]);
1116                          ) UNTIL CHECK(0,P) AND (M OR N=='2' OR N=='3');
1117                        )
1118                  ' ': I=TRUE;
1119                 '¥N': P=-1,I=TRUE;
1120     )
1121   ) UNTIL I;
1122 END(P);
1123
1124 // *INPUT YES OR NO****************
1125 YESNO(X,Y,R)
1126 BEGIN
1127   VAR K,D;
1128   LOCATE(X,Y);
1129   PRINT("YES/NO");
1130   D=FALSE;
1131   REPEAT (
1132     LOCATE(X+(1-R)*4,Y);
1133     K=INKEY(1);
1134     CASE K OF (
1135       'Y': R=(D=TRUE);
1136       'N': R=1-(D=TRUE);
1137       '4' ,'6' ,
1138       '¥R','¥L',
1139       'Z' ,'X'  : R=1-R;
1140       ' ','¥N': D=TRUE;
1141     )
1142   ) UNTIL D;
1143 END(R);
1144
1145 // *WAIT****************************
1146 WAIT(T)
1147 BEGIN
1148   VAR I;
1149   FOR I=1 TO T*WLOOP ;
1150 END;
1151
1152 // *LDIR****************************
1153 LDIR(3)
1154 BEGIN
1155   CODE(0EDH,0B0H);  // LDIR
1156 END;
1157
1158 //
1159 // *DEMONSTRATION PROGRAM**********
1160 //
1161
1162 CONST PLOOP=800,KLOOP=1200;
1163
1164 DEMO()
1165 BEGIN
1166   DEMO1();
1167   DEMO2();
1168 END;
1169
1170 // *DEMONSTRATION PART 1***********
1171 DEMO1()
1172 ARRAY !X[168]=( 6,6,5,4,3,4,5,6,7,8,9,10,11,
1173                 9,9,9,8,8,7,7,6,5,4,3,2,1,1,
1174                 3,4,5,6,7,8,9,10,11,10,9,9,9,
1175                 10,11,12,13,14,14,14,13,12,
1176                 12,13,14,15,16,16,17,18,
1177                 19,19,19,19,20,21,
1178                 6,7,8,9,10,11,12,13,14,14,15,15,
1179                 14,13,12,12,12,13,14,14,14,
1180                 13,12,11,10,9,8,7,7,7,8,9,10,11,
1181                 16,17,24,23,22,21,21,20,20,
1182                 19,19,19,20,21,22,23,24,
1183                 25,26,27,27,27,28,28,28,27,
1184                 26,25,24,23,24,24,
1185                 25,26,27,28,29,30,
1186                 29,30,31,32,33,34,35,36,37,37,38,38,
1187                 37,36,35,35,35,36,37,37,37,
1188                 36,35,34,33,32,31,30,30,30,
1189                 31,32,33,34 ),
1190       !Y[168]=( 1,2,3,3,2,1,1,1,1,1,2,2,1,
1191                 3,4,5,6,7,8,9,10,11,11,10,10,
1192                 9,8,7,7,7,7,7,6,6,6,6,7,8,9,10,
1193                 11,11,11,10,9,8,7,6,6,
1194                 7,8,8,8,7,6,6,7,8,9,10,11,11,10,
1195                 23,23,23,22,21,20,19,18,17,16,
1196                 15,14,13,14,15,16,17,18,19,
1197                 20,21,22,23,23,23,22,22,
1198                 21,20,19,20,21,21,21,
1199                 18,18,13,13,14,15,16,17,18,
1200                 19,20,21,22,23,23,23,23,22,21,
1201                 20,19,18,17,16,15,14,13,13,13,
1202                 14,15,16,17,17,17,17,17,16,
1203                 23,23,23,22,21,20,19,18,17,16,
1204                 15,14,13,14,15,16,17,18,19,
1205                 20,21,22,23,23,23,22,22,
1206                 21,20,19,20,21,21,21 );
1207 BEGIN
1208   VAR I;
1209   PRINT("¥C");
1210   FOR I=0 TO 168 (
1211     LOCATE(X[I],Y[I]);
1212     PRINT("#","¥L");
1213     PAUSE(1);
1214     PRINT(CHR$(7BH));
1215   )
1216   WAIT(10);
1217   PRINT(" ");
1218 END;
1219
1220 // *DEMONSTRATION PART 2***********
1221 DEMO2()
1222   ARRAY !X[6]=( 3,9,15,21,27,33,35 ),
1223         !Y[27]=( 2,2,2,2,2,2,
1224                  2,3,5,8,12,17,
1225                  14,12,11,11,12,14,
1226                  17,16,16,17,17,17,
1227                  17,17,17,17 );
1228 BEGIN
1229   VAR I,J,L,P;
1230   PRINT("¥C");
1231   FOR I=0 TO 6 (
1232     LOCATE(X[I],2);
1233     DCHRPRINT(I);
1234     BEEP();
1235     WAIT(3);
1236   )
1237   FOR I=7 TO 27
1238     FOR J=0 TO 6 (
1239       P=I-J;
1240       IF Y[P]>Y[P-1]
1241         THEN FOR L=Y[P-1] TO Y[P]-1 (
1242                LOCATE(X[J],L);
1243                PRINT(SPC$(J<5?5:1));
1244              )
1245       ELSE IF Y[P]<Y[P-1]
1246         THEN FOR L=Y[P]+5 TO Y[P-1]+4 (
1247                LOCATE(X[J],L);
1248                PRINT(SPC$(J<5?5:1));
1249              )
1250       LOCATE(X[J],Y[P]);
1251       DCHRPRINT(J);
1252     )
1253   WAIT(5);
1254   FOR I=17 TO 22 (
1255     LOCATE(3,I);
1256     PRINT(SPC$(33));
1257   )
1258   FOR I=0 TO 6 (
1259     LOCATE(X[I],2);
1260     DCHRPRINT(I);
1261   )
1262   DMSG2(0,9,"AN EXCITING CARD GAME FOR S-OS & SLANG!");
1263   BEEP();
1264   WAIT(2);
1265   LOCATE(4,0);
1266   PRINT("<< HIT ANY KEY TO GAME START >>");
1267   OVER=0;
1268   REPEAT (
1269     DMSG(2,12,"¥"DOBON!¥"ﾊ､ﾃﾓﾁﾉ ｶｰﾄﾞｦ ﾄﾞﾝﾄﾞﾝ ﾍﾗｼﾃｲｷ､");
1270     DMSG(2,14,"ﾊﾞ ﾆ ﾃﾞﾃｲﾙ ｶｰﾄﾞﾉ ｽｳｼﾞｶﾞ､ﾃﾓﾁﾉ ｶｰﾄﾞﾉ");
1271     DMSG(1,16,"ｽｳｼﾞﾉ ｺﾞｳｹｲﾄ ｵﾅｼﾞﾆ ﾅｯﾀﾗ ｱｶﾞﾙ ｹﾞｰﾑﾃﾞｽ｡");
1272     KWAIT(8);
1273     EXPCLEAR();
1274     DMSG(1,11," ｻｲｼｮﾆ ｶｰﾄﾞｦ 4ﾏｲｽﾞﾂ ｸﾊﾞﾘﾏｽ｡");
1275     DMSG(1,13," ﾔﾏ(PILE)ｶﾗ ｶｰﾄﾞｦ 1ﾏｲ ﾒｸﾘ､ｵﾔｶﾗ ｹﾞｰﾑｦ");
1276     DMSG(1,15,"ﾊｼﾞﾒﾏｽ｡");
1277     DMSG(1,17," ｼﾞﾌﾞﾝﾉ ﾊﾞﾝﾆ ﾅｯﾀﾗ､ﾊﾞﾆ ﾃﾞﾃｲﾙ ｶｰﾄﾞﾄ､");
1278     DMSG(1,19,"ｽｳｼﾞｶ ｽｰﾄｶﾞ ｵﾅｼﾞ ｶｰﾄﾞｦ ﾀﾞｼﾏｽ｡");
1279     DMSG(1,21," ﾀﾞｾﾙ ｶｰﾄﾞｶﾞ ﾅｲﾄｷﾊ､PILEｶﾗ 1ﾏｲ ﾋｷﾏｽ｡");
1280     DMSG(1,23,"ﾋｲﾀ ｶｰﾄﾞｦ ｽｸﾞ ﾀﾞｽｺﾄﾓ ﾃﾞｷﾏｽ｡");
1281     KWAIT(8);
1282     EXPCLEAR();
1283     DMSG(1,11," ｶｰﾄﾞﾉ ﾅｶﾆﾊ 5ｼｭﾙｲﾉ ﾔｸﾌﾀﾞｶﾞ ｱｯﾃ､ｿﾚｿﾞﾚ");
1284     DMSG(1,13,"ﾂｷﾞﾉ ﾖｳﾅ ﾊﾀﾗｷｦ ｼﾏｽ｡");
1285     DMSG(1,15,"* A(SKIP):ﾂｷﾞﾉ ﾋﾄｿ ｼﾞｭﾝﾊﾞﾝｦ ﾄﾊﾞｼﾏｽ｡");
1286     DMSG(1,17,"* 8(ALMIGHTY):ｽｰﾄｦ ｼﾃｲﾃﾞｷﾏｽ｡");
1287     DMSG(1,19,"* 9(REVERSE):ｷﾞｬｸﾏﾜﾘﾆ ﾅﾘﾏｽ｡");
1288     DMSG(1,21,"* 2,3(DRAW 2/3):ﾂｷﾞﾉ ﾋﾄﾆ 2ﾏｲ ﾏﾀﾊ 3ﾏｲ");
1289     DMSG(17,23,"ｶｰﾄﾞｦ ﾋｶｾﾏｽ｡");
1290     KWAIT(8);
1291     EXPCLEAR();
1292     DMSG(1,11," DRAW 2/3ｦ ﾀﾞｻﾚﾀﾄｷﾆ､ﾓｼ ｽｳｼﾞｶ ｽｰﾄﾉ ｵﾅｼﾞ");
1293     DMSG(1,13,"DRAW 2/3ｦ ﾓｯﾃｲﾚﾊﾞ､ｶｻﾈﾃ ﾀﾞｽｺﾄｶﾞ ﾃﾞｷﾏｽ｡");
1294     DMSG(1,15,"(EX.) A => B => C ﾉ ｼﾞｭﾝﾆ ﾏﾜｯﾃｲﾙ ﾄｷﾆ､");
1295     DMSG(7,17,"Aｶﾞ SPADEﾉ 3 ｦ ﾀﾞｼﾏｼﾀ｡");
1296     DMSG(7,19,"ｺｺﾃﾞ Bｶﾞ SPADEﾉ 2ｦ ﾓｯﾃｲﾚﾊﾞ");
1297     DMSG(7,21,"DRAW 5ﾄｼﾃ､ﾏﾀ ﾎｶﾉ ｽｰﾄﾉ 3ｦ ﾓｯﾃｲﾚﾊﾞ");
1298     DMSG(7,23,"DRAW 6ﾄｼﾃ､Cﾆ ﾋｶｾﾙ ｺﾄｶﾞ ﾃﾞｷﾏｽ｡");
1299     KWAIT(8);
1300     EXPCLEAR();
1301     DMSG(1,11," ｺｳｼﾃ ｶｰﾄﾞｦ ﾍﾗｼﾃｲｯﾃ､ﾎｶﾉ ﾋﾄﾉ ﾀﾞｼﾀ ｶｰﾄﾞﾉ");
1302     DMSG(1,13,"ｽｳｼﾞｶﾞ､ｼﾞﾌﾞﾝﾉ ﾃﾌﾀﾞﾉ ｽｳｼﾞ(J ﾊ 11,Q ﾊ ");
1303     DMSG(1,15,"12,K ﾊ 13ﾄｼﾃ ｶｿﾞｴﾏｽ｡)ﾉ ｺﾞｳｹｲﾄ ｵﾅｼﾞﾆ");
1304     DMSG(1,17,"ﾅｯﾀﾗ､｢ﾄﾞﾎﾞﾝ!｣ﾄ ｲｯﾃ ｱｶﾞﾘﾏｽ｡");
1305     DMSG(1,19," ﾀﾀﾞｼ､ｺﾉﾄｷﾆ､ｶｰﾄﾞｦ ﾀﾞｼﾀ ﾋﾄﾉ ﾃﾌﾀﾞﾉ ｽｳｼﾞﾉ");
1306     DMSG(1,21,"ｺﾞｳｹｲﾓ ｵﾅｼﾞﾀﾞｯﾀﾗ､｢ﾄﾞﾎﾞﾝｶﾞｴｼ!!｣ﾄ ｲｯﾃ､");
1307     DMSG(1,23,"｢ﾄﾞﾎﾞﾝ｣ｼﾀ ﾋﾄﾉ ﾏｹﾆ ﾅﾘﾏｽ｡");
1308     KWAIT(8);
1309     EXPCLEAR();
1310     DMSG(1,11," ｢ﾄﾞﾎﾞﾝｶﾞｴｼ｣ｻﾚﾅｶｯﾀ ﾄｷﾊ､｢ﾄﾞﾎﾞﾝ｣ｼﾀ ﾋﾄｶﾞ");
1311     DMSG(1,13,"ｶﾁ､ｶｰﾄﾞｦ ﾊﾞﾆ ﾀﾞｼﾀ ﾋﾄｶﾞ ﾏｹ｡");
1312     DMSG(1,15," ｢ﾄﾞﾎﾞﾝｶﾞｴｼ｣ｻﾚﾀ ﾄｷﾊ､｢ﾄﾞﾎﾞﾝｶﾞｴｼ｣ｼﾀ ﾋﾄｶﾞ");
1313     DMSG(1,17,"ｶﾁ､｢ﾄﾞﾎﾞﾝ｣ｼﾀ ﾋﾄｶﾞ ﾏｹ｡");
1314     DMSG(1,19," ﾂｷﾞﾉ ｹﾞｰﾑﾃﾞﾊ､ ｶｯﾀ ﾋﾄｶﾞ ｵﾔﾆ ﾅﾘ､ｶｰﾄﾞｦ");
1315     DMSG(1,21,"1ﾏｲ ｽｸﾅｸｼﾃ(ﾀﾀﾞｼ 1ﾏｲﾅﾗ ｿﾉﾏﾏ)､ﾏﾀ ﾏｹﾀ ﾋﾄﾊ");
1316     DMSG(1,23,"1ﾏｲ ﾌﾔｼﾃ(10ﾏｲﾅﾗ ｿﾉﾏﾏ) ﾊｼﾞﾒﾏｽ｡");
1317     KWAIT(8);
1318     EXPCLEAR();
1319   ) UNTIL OVER;
1320 END;
1321
1322 // *BIG CHARACTER DISPLAY(DEMO P2)*
1323 DCHRPRINT(CH)
1324 BEGIN
1325   CASE CH OF (
1326     0  : PRINT(CHR$(7D7BH),CHR$(7D7BH)," ¥D¥L¥L¥L¥L¥L",
1327                CHR$(7BH),"   ",CHR$(7BH),"¥D¥L¥L¥L¥L¥L",
1328                CHR$(7BH),"   ",CHR$(7BH),"¥D¥L¥L¥L¥L¥L",
1329                CHR$(7BH),"   ",CHR$(7BH),"¥D¥L¥L¥L¥L¥L",
1330                CHR$(7D7BH),CHR$(7D7BH)," ");
1331     1,3: PRINT(" ",CHR$(7D7BH),CHR$(7BH)," ¥D¥L¥L¥L¥L¥L",
1332                CHR$(7BH),"   ",CHR$(7BH),"¥D¥L¥L¥L¥L¥L",
1333                CHR$(7BH),"   ",CHR$(7BH),"¥D¥L¥L¥L¥L¥L",
1334                CHR$(7BH),"   ",CHR$(7BH),"¥D¥L¥L¥L¥L¥L",
1335                " ",CHR$(7D7BH),CHR$(7BH)," ");
1336     2  : PRINT(CHR$(7D7BH),CHR$(7D7BH)," ¥D¥L¥L¥L¥L¥L",
1337                CHR$(7BH),"   ",CHR$(7BH),"¥D¥L¥L¥L¥L¥L",
1338                CHR$(7D7BH),CHR$(7D7BH)," ¥D¥L¥L¥L¥L¥L",
1339                CHR$(7BH),"   ",CHR$(7BH),"¥D¥L¥L¥L¥L¥L",
1340                CHR$(7D7BH),CHR$(7D7BH)," ");
1341     4  : PRINT(CHR$(7BH),"   ",CHR$(7BH),"¥D¥L¥L¥L¥L¥L",
1342                CHR$(7D7BH),"  ",CHR$(7BH),"¥D¥L¥L¥L¥L¥L",
1343                CHR$(7BH)," ",CHR$(7BH)," ",CHR$(7BH),"¥D¥L¥L¥L¥L¥L",
1344                CHR$(7BH),"  ",CHR$(7D7BH),"¥D¥L¥L¥L¥L¥L",
1345                CHR$(7BH),"   ",CHR$(7BH));
1346     5,6: PRINT(CHR$(7BH),"¥D¥L",
1347                CHR$(7BH),"¥D¥L",
1348                CHR$(7BH),"¥D¥L ¥D¥L",
1349                CHR$(7BH));
1350   )
1351 END;
1352
1353 // *DISPLAY MESSAGE WITH CURSOR****
1354 DMSG(X,Y,ADR)
1355 BEGIN
1356   VAR C;
1357   IF OVER THEN RETURN;
1358   LOCATE(X,Y);
1359   WHILE C=MEM[ADR++] (
1360     PRINT(CHR$(7BH),"¥L");
1361     PAUSE(1);
1362     PRINT(CHR$(C));
1363     IF OVER=INKEY(0) THEN EXIT;
1364   )
1365 END;
1366
1367 // *DISPLAY MESSAGE WITH CURSOR****
1368 // *(IGNORE KEYIN)*****************
1369 DMSG2(X,Y,ADR)
1370 BEGIN
1371   VAR C;
1372   LOCATE(X,Y);
1373   WHILE C=MEM[ADR++] (
1374     PRINT(CHR$(7BH),"¥L");
1375     PAUSE(1);
1376     PRINT(CHR$(C));
1377   )
1378 END;
1379
1380 // *EXPLAIN CLEAR******************
1381 EXPCLEAR()
1382 BEGIN
1383   VAR I;
1384   IF OVER THEN RETURN;
1385   FOR I=10 TO 23 (
1386     LOCATE(0,I);
1387     PRINT(SPC$(40));
1388   )
```

▶いままでOh！Xを読んでいて，２浪するなんてどんな人なんだろうと思っていたが，なんのことはない，自分のことだったりする。ああ……。　田中　一成(19)京都府

```
1389 END;
1390
1391 // *WAIT(KEYIN=>OVER)****************
1392 KWAIT(T)
1393 BEGIN
1394   VAR I;
1395   IF OVER THEN RETURN;
1396   FOR I=1 TO T*KLOOP
1397     IF OVER=INKEY(0) THEN EXIT;
1398 END;
1399
1400 // *PAUSE*****************************
1401 PAUSE(T)
1402 BEGIN
```

```
1403   VAR I;
1404   FOR I=1 TO T*PLOOP ;
1405 END;
1406
1407 // *PROGRAM END*********************
1408 //
1409 // BASIC PROGRAM : '90 SPRING VACATION
1410 // SLANG PROGRAM : '90/8/13
1411 // THANKS TO : MZ-2531,MZ-1D22,
1412 //             S-OS"SWORD",WINER,SLANG,
1413 //             Fuzzy BASIC,
1414 //             OH!MZ '86 NOV. ASCII SAVE PROGRAM,
1415 //                                          ETC.
1416 //
```

全機種共通システムインデックス

■85年6月号
- 序論　共通化の試み
- 第1部　S-OS"MACE"
- 第2部　Lisp-85インタプリタ
- 第3部　チェックサムプログラム

■85年7月号
- 第4部　マシン語プログラム開発入門
- 第5部　エディタアセンブラZEDA
- 第6部　デバッグツールZAID

■85年8月号
- 第7部　ゲーム開発パッケージBEMS
- 第8部　ソースジェネレータZING

■85年9月号
- インタラプト　S-OS番外地
- 第9部　マシン語入力ツールMACINTO-S
- 第10部　Lisp-85入門(1)

■85年10月号
- 第11部　仮想マシンCAP-X85
- 連載　Lisp-85入門(2)

■85年11月号
- 連載　Lisp-85入門(3)

■85年12月号
- 第12部　Prolog-85発表

■86年1月号
- 第13部　リロケータブルのお話
- 第14部　FM音源サウンドエディタ

■86年2月号
- 第15部　S-OS"SWORD"
- 第16部　Prolog-85入門(1)

■86年3月号
- 第17部　magiFORTH発表
- 連載　Prolog-85入門(2)

■86年4月号
- 第18部　思考ゲームJEWEL
- 第19部　LIFE GAME
- 連載　基礎からのmagiFORTH
- 連載　Prolog-85入門(3)

■86年5月号
- 第20部　スクリーンエディタE-MATE
- 連載　実戦演習magiFORTH

■86年6月号
- 第21部　Z80TRACER
- 第22部　magiFORTH TRACER
- 第23部　ディスクダンプ&エディタ
- 第24部　"SWORD" 2000 QD
- 連載　対話で学ぶ magiFORTH
- 特別付録　PC-8801版S-OS"SWORD"

■86年7月号
- 第25部　FM音源ミュージックシステム
- 付録　FM音源ボードの製作
- 連載　計算力アップのmagiFORTH
- 特別付録　SMC-777版 S-OS"SWORD"

■86年8月号
- 第26部　対局五目並べ
- 第27部　MZ-2500版 S-OS"SWORD"

■86年9月号
- 第28部　FuzzyBASIC 発表
- 連載　明日に向かって magiFORTH

■86年10月号
- 第29部　ちょっと便利な拡張プログラム
- 第30部　ディスクモニタ DREAM
- 第31部　FuzzyBASIC 料理法<1>

■86年11月号
- 第32部　パズルゲーム HOTTAN
- 第33部　MAZE in MAZE
- 連載　FuzzyBASIC 料理法<2>

■86年12月号
- 第34部　CASL & COMET
- 連載　FuzzyBASIC 料理法<3>

■87年1月号
- 第35部　マシン語入力ツールMACINTO-C
- 連載　FuzzyBASIC 料理法<4>

■87年2月号
- 第36部　アドベンチャーゲーム MARMALADE
- 第37部　テキアベ作成ツール CONTEX

■87年3月号
- 第38部　魔法使いはアニメがお好き
- 第39部　アニメーションツール MAGE
- 付録　"SWORD" 再掲載と MAGIC の標準化

■87年4月号
- 第40部　INVADER GAME
- 第41部　TANGERINE

■87年5月号
- 第42部　S-OS"SWORD" 変身セット
- 第43部　MZ-700用 "SWORD" を QD 対応に

■87年6月号
- インタラプト　コンパイラ物語
- 第44部　FuzzyBASIC コンパイラ
- 第45部　エディタアセンブラ ZEDA-3

■87年7月号
- 第46部　STORY MASTER

■87年8月号
- 第47部　パズルゲーム碁石拾い
- 第48部　漢字出力パッケージ JACKWRITE
- 特別付録　FM-7/77版 S-OS"SWORD"

■87年9月号
- 第49部　リロケータブル逆アセンブラ Inside-R
- 特別付録　PC-8001/8801 版 S-OS"SWORD"

■87年10月号
- 第50部　tiny CORE WARS
- 第51部　FuzzyBASIC コンパイラの拡張
- 第52部　X1turbo 版 S-OS"SWORD"

■87年11月号
- 序論　神話のなかのマイクロコンピュータ
- 付録　S-OS の仲間たち
- 第53部　もうひとつの FuzzyBASIC 入門
- 第54部　ファイルアロケータ&ローダ
- インタラプト　S-OS こちら集中治療室
- 第55部　BACK GAMMON

■87年12月号
- 第56部　タートルグラフィックパッケージTURTLE
- 第57部　X1turbo 版 "SWORD" アフターケア ラインプリントルーチン
- 特別付録　PASOPIA7 版 S-OS"SWORD"

■88年1月号
- 第58部　FuzzyBASIC コンパイラ・奥村版
- 付録　石上版コンパイラ拡張部の修正

■88年2月号
- 第59部　シューティングゲーム ELFES

■88年3月号
- 第60部　構造型コンパイラ言語 SLANG

■88年4月号
- 第61部　デバッギングツール TRADE
- 第62部　シミュレーションウォーゲーム WALRUS

■88年5月号
- 第63部　シューティングゲーム ELFES II
- 第64部　地底最大の作戦

■88年6月号
- 第65部　構造化言語 SLANG 入門(1)
- 第66部　Lisp-85 用 NAMPA シミュレーション

■88年7月号
- 第67部　マルチウィンドウドライバ MW-1
- 連載　構造化言語 SLANG 入門(2)

■88年8月号
- 第68部　マルチウィンドウエディタ WINER

■88年9月号
- 第69部　超小型エディタ TED-750
- 第70部　アフターケア WINER の拡張

■88年10月号
- 第71部　SLANG 用ファイル入出力ライブラリ
- 第72部　シューティングゲーム MANKAI

■88年11月号
- 第73部　シューティングゲーム ELFES IV

■88年12月号
- 第74部　ソースジェネレータ SOURCERY

■89年1月号
- 第75部　パズルゲーム LAST ONE
- 第76部　ブロックゲーム FLICK

■89年2月号
- 第77部　高速エディタアセンブラ REDA
- 特別付録　X1版 S-OS"SWORD"<再掲載>

■89年3月号
- 第78部　Z80用浮動小数点演算パッケージSOROBAN

■89年4月号
- 第79部　SLANG 用実数演算ライブラリ

■89年5月号
- 第80部　ソースジェネレータ RING

■89年6月号
- 第81部　超小型コンパイラTTC

■89年7月号
- 第82部　TTC用パズルゲーム　TICBAN

■89年8月号
- 第83部　CP/M用ファイルコンバータ

■89年9月号
- 第84部　生物進化シミュレーションBUGS

■89年10月号
- 第85部　小型インタプリタ言語TTI

■89年11月号
- 第86部　TTI用パズルゲーム PUSH BON!

■89年12月号
- 第87部　SLANG用リダイレクションライブラリ DIO. LIB

■90年1月号
- 第88部　SLANG用ゲームWORM KUN
- 特別付録　再掲載SLANGコンパイラ

■90年2月号
- 第89部　超小型コンパイラTTC++

■90年3月号
- 第90部　超多機能アセンブラOHM-Z80

■90年4月号
- 第91部　ファジィコンピュータシミュレーションI-MY

■90年5月号
- 第92部　インタプリタ言語STACK

■90年6月号
- 第93部　リロケータブルフォーマットの取り決め
- 第94部　STACK用ゲーム SQUASH！
- 第95部　X68000対応S-OS"SWORD"
- 特別付録　PC-286対応S-OS"SWORD"

■90年7月号
- 第96部　リロケータブルアセンブラWZD

■90年8月号
- 第97部　リンカWLK

■90年9月号
- 第98部　BILLIARDS

■90年10月号
- 第99部　ライブラリアンWLB

■90年11月号
- 第100部　タブコード対応エディタEDC-T

■90年12月号
- 第101部　STACKコンパイラ

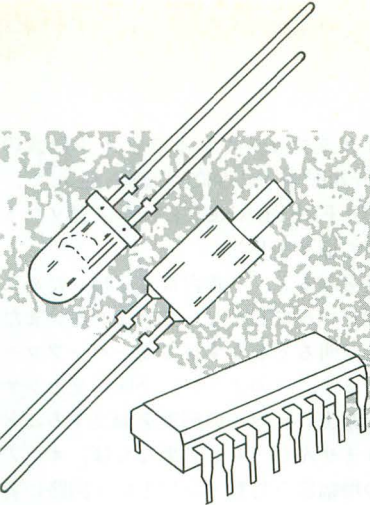
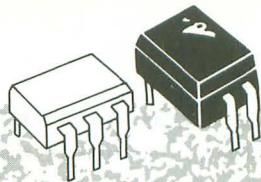
■91年1月号
- 第102部　ブロックアクションゲーム COLUMNS

■91年2月号
- 第103部　ダイスゲームKISMET

■91年3月号
- 第104部　アクションゲームMUD BALLIN'

＊以上のアプリケーションは，基本システムである S-OS"MACE"または S-OS"SWORD" がないと動作しませんのでご注意ください。

ハードウェア工作入門《10》

センサー回路その4

Misawa Kazuhiko
三沢 和彦

センサーに関する理論編を経て前回はアルコールセンサーと光センサーの2つの製作を終えました。いよいよ今回でセンサー回路は終了です。製作した回路を利用して簡単な応用プログラムを組んでみましょう。

センサー回路の工作はうまくできましたか。メインのA/Dコンバータさえうまく動作していたら，センサー部分のほうは特に問題はないだろうと思います。では，今回はこれらのセンサー回路を使って，簡単なプログラムを組んでみましょう。なお，前回の記事でちょっとした手違いから，文中で指示されているリスト1が割愛されてしまいました。申し訳ありません。リスト0として掲載しておきます。

デジタル電圧計表示プログラム

まずはA/Dコンバータの応用として最も基本的な電圧計をプログラムしてみます。これは，センサー回路に限らず，A/Dコンバータを使ううえで最初にプログラムしてもよかったのですが，センサー回路を製作して初めて測定器として実用になるので，今回までとっておきました。

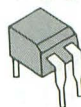
主要な部分はリスト1に挙げてあります。プログラムを打ち込んだあと，A/Dコンバータに簡易ジョイスティックをつないでからRUNさせます。簡易ジョイスティックを工作していない人は実際にセンサー回路をつなぐところまで読み飛ばしてかまいません。

さて，プログラムがスタートすると画面に指針式のメーターと値の数字が表示されます。誤差などのふらつきで，指針も数字もふらついていると思いますが，問題ありません。A/DコンバータがX68000本体からはずれていると，値は5Vに振り切れたままになっているはずですのでチェックしてください。簡易ジョイスティックがつながっていて，正常に動作しているときはボリュームツマミを回すとそれに反応して指針が動きます。

それでは，リストに従ってプログラムを解説していきましょう。A/Dコンバータからの読み込みルーチンはすでにこれまでのプログラムに何度も出てきているread関数を使っています。10000行以降がread関数になっていて，今回のリストすべてに共通のルーチンになっていますので，リスト2～4では省略していますが，やはり10000行以降も入力してください。外部関数のバージョンアップをしている人はread関数の代わりにadread関数を使えば，この部分のread関数のルーチンは打ち込む必要がありません。

このプログラムの本質的な部分は5000行からで，グラフィック画面上に電圧計の目盛りを表示させるscaleルーチンとA/Dコンバータから読み取ってきた値を表示するmeterルーチンとからなっています。scaleルーチンは扇型の帯を描くbeltルーチンを3回呼び出しているだけで，これによって，ブルーゾーン，イエローゾーン，レッドゾーンを描きます。beltルーチンの入力パラメータは開始角度，終了角度，及びカラーになっていて，標準ではイエローゾーンがフルスケールの70%～90%，レッドゾーンが90%～100%の範囲を示すようにパラメータを指定しています。

meterルーチンで表示される指針はグラフィックプレーン0上に描いているので，目盛りと無関係に書いたり消したりすることができます。メインループに入って，read関数でA/Dコンバータからの出力を読み込むと，その値をフルスケール255で割り算して規格化した値をmeterルーチンに渡し，meterルーチン内で指針を描いてくれます。また，同時にデジタル表示も画面中央に現れます。

これらの基本ルーチンも今月のリストすべてに共通ですから，やはり5000行以降も必ず入力してください。では，これらのルーチンを応用して，次に飲酒チェッカーと照度計とをプログラムしてみたいと思います。

飲酒チェッカーのプログラミング

A/Dコンバータにアルコールセンサーをつないだときの様子をここでもう一度復習しておきましょう。センサーを接続したA/Dコンバータボードの出力はまず0まで下がってからだんだん増えていき，40ぐらいまで上がったら今度はゆっくり下がっていきます。そして最終的には30ぐらいで落ち着きます。アルコール蒸気にアルコールセンサーの素子面を近づけてみると，画面の値が急激に上がり，また遠ざけると値が元に戻っていきます。

そこで，飲酒チェッカーの動作としては，

1) アルコールセンサーのウォーミングアップが終わって，測定スタンバイになったことを知らせる。
2) 吐く息のアルコール含有量を測定して，表示する。
3) 車を運転したらどうなるかのメッセージを表示する。

という順番に進めていこうと思います。

サンプルプログラムはリスト2を見てください。主要な部分はリスト1と重複しています。リスト2では省略していますが，

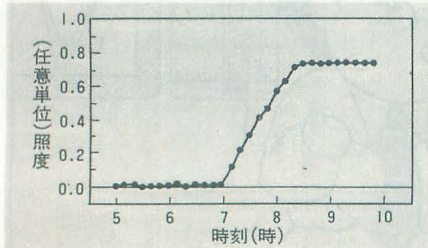
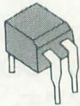
リスト1の5000行以降が必要です。

問題は，1000行からの飲酒チェックの判定のアルゴリズムですので，そこを中心に説明していきましょう。まず，測定スタンバイであるかどうかは，フルスケールの20%以下に収まっていればよいということにします。

プログラムをRUNさせた直後だけは，ウォーミングアップでいったん出力が上がっていくのを確認してから動作に入るようにしています。すでにウォーミングアップが終わってからRUNさせるとずっと待ち続けてしまうので，そのときは試しに一度アルコールセンサーに息を吐きかけてみるとよいでしょう。測定スタンバイになると，表示が「しばらくお待ちください」から，「息を吹き込んでください」に変わりますので，実際に息を吐き込んでみます。

プログラムでは，breathルーチンで，測定値のピーク値を検出するようになっていて，息を吐き込むのを止めるとそのピーク値をcheckルーチンに渡して飲酒の判定に移ります。checkルーチンでは測定値がフルスケールの70%以上になると，完全に飲酒運転の状態になるので，ブザー音とともに画面に警告メッセージが点滅して知らせます。フルスケールの40%以上では，酒気帯び運転の状態と判断して，ビープ音とともに警告メッセージが表示されます。それ以下の場合は一応飲酒については問題なしと判断します。それぞれの判定値について，もし皆さんのほうで実際にテストしてみた結果，値を変えたいという場合には，初期設定を変えてください。

さて，今回製作に用いたアルコールセンサーと同様のものが警察の飲酒運転取り締まりに使われているかどうかは確かではありませんが，この飲酒チェッカーで実際にテストしてみた結果，フーッと息を吹きかけるときとハーッと息を吐きかけるときとではハーッと息を吐くほうが測定値が低いことがわかかりました。

これまでは，ハーッと吐くと胃の奥のほうから空気が出てくるのに対し，フーッと息を吹くほうは口の中からしか空気が出てこないので，ハーッのほうがアルコール含有量が多いと考えられてきたようですが，それは迷信だったようです。だからといって，今回の飲酒チェッカーでどうしたら警察をごまかせるかなんて練習しないように。飲んだら絶対運転してはいけません。

照度計とおはようアラーム

照度計といっても，リスト1の電圧計に光センサーを接続するだけです。リスト3では光センサーからの電圧値を10回平均していますが，これは，螢光燈などでは50Hzで振動しているので，その変動をならしてしまうためです。

5000行以降は共通ルーチンで，1000行からはリスト1のvol関数の代わりにlux関数に入れ替えただけ，また，メインルーチンもvol関数の呼び出しの部分をlux関数に変更しただけです。さて，先月述べたように，オペアンプLM358の出力は電源電圧が＋5Vのときはフルスケールで＋3.8V程度なので，X68000の電圧計につないでも5Vフルスケールまで振り切れません。フォトダイオードに懐中電灯を近づけたときに3.7～3.8Vを指していれば，OKです。

では最後に，朝になって日光が部屋の中に差し込んできたら，アラームを鳴らすプログラムを組んでみましょう。その予備段階として，まず朝の各時刻に部屋に差し込んでくる光の量を記録します。プログラムはリスト4のように500行以降のmeasureルーチンを照度計に追加する形で十分です。実際に動作させるときはリスト4を入力したあとにリスト3の500行以降と全リスト共通の1000行以降を加えて入力すると間違いないでしょう。

このmeasureルーチンでは，TIME$関数で時刻を監視しておき，5：00から9：50まで10分おきに計30回照度計の読みをファイルに記録していきます。ファイルは“luxval.dat”と“luxchr.dat”の2種類で，“luxval.dat”のほうには8バイトfloat形式で記録し，“luxchr.dat”のほうには値を文字列に直して，テキストファイルの形式で記録していきます。

この記録のための配列変数がプログラムの最初のほうで宣言されていますので，忘れないように。そして，皆さんの必要に応じてどちらかのファイルを参照してください。

さて，私の部屋の窓際に光センサーを置いておいたときの結果を図1に示します。グラフは横軸が時間で縦軸が照度計の読みです。グラフはテキストファイル形式の“luxchr.dat”をPC-9801で読み込んで，LOTUS FREELANCEという作表ソフトウェアで描きました。

図1

結果をみると，5時台はまだ暗いためにノイズレベルにしかなっていません。また9時台は明るくなりすぎてオーバーフローしています。このように，今回の光センサーでは幅広いレベルの照度を測定することはできませんが（もし必要ならば，オペアンプの増幅器を対数アンプという回路に変更すればできます），7時台では時間とともに明るさが増していく様子がわかります。

この結果を利用して，おはようアラームのプログラムを組むには，ずっと照度計で窓際の明るさを監視しておいて，目的の時刻に対応した照度に達したらアラーム動作に移るようにすればよいでしょう。サンプルプログラムをリスト5に挙げます。実際の動作にはリスト5に加えて，1000行からのlux関数と10000行からのread関数を入力してください。

さて，このおはようアラームの最大の欠点は，その日の天候によって明るさにバラツキがあるので正確に時刻を再現しない，ということと，季節によって日の出と日の入りの時刻が変わっていくので，長期間は使えないということです。まあ時刻に正確になったのは現代人であって，昔の人々のように日の出とともに活動を開始し，日の入りとともに就寝するという生活形態に変えれば，便利なアラームとなるでしょう。プログラム中のalarm関数の中身を自由に書き換えてみてください。

*

今月はセンサー回路を応用した実用(？)プログラムをいくつか組んでみました。アルコールセンサーはホビーユースよりの飲酒チェッカーとして応用できましたが，光センサーのほうは照度計といった専門の計測器よりのプログラムとなってしまいました。特に，時刻と照度との関係を計測してコンピュータにデータとして取り込み，それをやはりコンピュータ上でグラフ化するという手順は，自然科学・産業技術分野でよく用いられている自動計測処理のきわめて簡単な例となっています。

X68000にもこのような専門的な使い方に耐えられるだけのポテンシャルは十分持ち合わせているのです。さらに来月からは，視点を変えてメカトロニクスへの応用を取り上げてみたいと思っています。これまでとはまたひと味違ったハードウェアが登場することでしょう。

リスト0　A/Dコンバータ読み取りサンプル

```
 10 /* save "d:¥basic¥adconv2.bas
 20 /*
 30 /*
 40 /*2チャンネル入力・8ビットADコンバータ
 50 /*      読み取りサンプルプログラム
 60 /*
 70 /*       1991.1.4  K. Misawa
 80 /*
 90 int cs=0,di=0
100 int xx,yy
110 /*
120 /*読み取りチャンネルCH0
130 int ch=0
140 /*
150 screen 1,1,1,1 : window(0,0,511,511)
160 apage(0) : vpage(15)
170 ioout(&B0)
180 /*
190 while 1
200   for xx=0 to 511
210     yy=adread(ch)
220     pset(xx,yy*2,15)
230     print yy
240   next
250   wipe()
260 endwhile
270 end
```

リスト1　電圧計表示プログラム

```
   10 /* save "d:¥basic¥digivol.bas
   20 /* save@"d:¥basic¥digivol.doc
   30 /*
   40 /*8ビットADコンバータ
   50 /*        電圧計表示プログラム
   60 /*
   70 /*     1991.2.3  K. Misawa
   80 /*
   90 int cs=0,di=0
  100 int xx,yy
  110 int rad=300,ox=250,oy=500
  120 float rd,elp=256#/360#,fs=5#
  130 rd=pi(1)/180#
  140 /*
  150 float v,vt=0
  160 /*
  170 /*読み取りチャンネルCH0
  180 int ch=0
  190 /*
  200 screen 1,1,1,1 : window(0,0,511,511)
  210 console ,,0
  220 vpage(15)
  230 ioout(&B0)
  240 scale()
  250 apage(0)
  260 while 1
  270   v=vol()
  280   meter(v)
  290 endwhile
  300 end
 1000 /*
 1010 /*電圧測定ルーチン
 1020 /*(引数)なし
 1030 /*(戻り値)測定値フルスケール0－1
 1040 /*
 1050 func float vol()
 1060   float v,vt=0
 1070   v=read(ch)/255#
 1080   return(v)
 1090 endfunc
 5000 /*
 5010 /*電圧計目盛り表示
 5020 /*
 5030 func scale()
 5040   apage(1)
 5050   belt(45,135,2)
 5060   belt(135-81,135-63,13)
 5070   belt(135-90,135-81,5)
 5080 endfunc
 5090 /*
 5100 /*目盛りベルト
 5110 /*(引数1)開始角度
 5120 /*(引数2)終了角度
 5130 /*(引数3)カラー
 5140 /*(戻り値)なし
 5150 /*
 5160 func belt(ts;int,te;int,c;int)
 5170   circle(ox,oy,rad,c,ts,te,360)
 5180   circle(ox,oy,rad*3/4,c,ts,te,360)
 5190   line(ox+rad*3/4*cos(ts*rd)*elp,oy-rad*3/4*sin(ts*rd),ox+
rad*cos(ts*rd)*elp,oy-rad*sin(ts*rd),c)
 5200   line(ox+rad*3/4*cos(te*rd)*elp,oy-rad*3/4*sin(te*rd),ox+
rad*cos(te*rd)*elp,oy-rad*sin(te*rd),c)
 5210   paint(ox+rad*7/8*cos((te+ts)*rd/2)*elp,oy-rad*7/8*sin((t
e+ts)*rd/2),c)
 5220 endfunc
 5230 /*
 5240 /*電圧計メーター表示
 5250 /*(引数)電圧値　規格化フルスケール0－1
 5260 /*(戻り値)なし
 5270 /*
 5280 func meter(v;float)
 5290   float tv
 5300   tv=135-90*v
 5310   color 7 : locate 28,20 : print using"#.####";v*fs
 5320   line(ox,oy,ox+rad*9/8*cos(tv*rd)*256#/360#,oy-rad*9/8*si
n(tv*rd),15)
 5330   line(ox,oy,ox+rad*9/8*cos(tv*rd)*256#/360#,oy-rad*9/8*si
n(tv*rd),0)
 5340 endfunc
10000 /*
10010 /*AD変換データリード
10020 /*(引数)入力チャンネル
10030 /*(戻り値)データ　5Vフルスケール
10040 /*                    0－255
10050 /*
10060 func int read(ch;int)
10070   int v=0
10080   ioout(&B1000000)
10090     start(1,ch)
10100     for iii=0 to 7
10110       clock(1)
10120       v=v*2+(ioinp() and 1)
10130     next
10140   ioout(&B0)
10150   return(v)
10160 endfunc
10170 /*
10180 /*AD変換スタート
10190 /*(引数1)ADC0832入力モード
10200 /*              1:単一入力
10210 /*           0:差動入力
10220 /*(引数2)入力チャンネル
10230 /*(戻り値)なし
10240 /*
10250 func start(sgl;int,sign;int)
10260   clock(1)
10270   clock(sgl)
10280   clock(sign)
10290 endfunc
10300 /*
10310 /*外部クロックルーチン
10320 /*(引数)ADC0832コマンド入力(DI)
10330 /*(戻り値)なし
10340 /*
10350 func clock(di;int)
10360   int stat
10370   stat=&B1000000+(1-di)*&B10000000
10380   ioout(stat+&B0)
10390   ioout(stat+&B10000)
10400   ioout(stat+&B0)
10410 endfunc
```

リスト2　飲酒チェッカープログラム

```
 10 /* save "d:¥basic¥alcohol.bas
 20 /* save@"d:¥basic¥alcohol.doc
 30 /*
 40 /*アルコールセンサー応用
 50 /*  飲酒チェッカープログラム
 60 /*
 70 /*     1991.2.7  K. Misawa
 80 /*
 90 int cs=0,di=0
100 /*
110 /*電圧計目盛りの半径、中心
120 /*
130 int rad=300,ox=250,oy=500
140 float rd,elp=256#/360#,fs=1#
150 rd=pi(1)/180#
160 /*
170 float v
180 /*
190 /*判定しきい値
200 /*
210 float thrval(2)={0.7#, 0.4#, 0.2#}
220 /*
230 /*警告メッセージ文
240 /*
250 dim str message(2)[60]= {"このまま運転したら、飲酒運転で免許取り消し確実！",
260                           "酒気おびがばれるから、運転しない方がよいでしょう。",
270                           "このままなら大丈夫。でも、運転は気を付けて。"}
280 /*
290 /*読み取りチャンネルCH0
300 /*
310 int ch=0
320 /*
330 /*イニシャライズ
340 /*
350 screen 1,1,1,1 : window(0,0,511,511)
360 console ,,0 : color 15
370 vpage(15)
380 ioout(&B0)
390 scale()
400 apage(0)
410 /*
420 color 6 : locate 20,4 : print "ウォーミングアップ中。"
430 warmflg=0
440 while warmflg=0
450   v=read(ch)/255#
460   meter(v)
470   if v>=thrval(2) then warmflg=1
480 endwhile
490 /*
500 /*メインルーチン
510 /*
520 while 1
530   cls
540   stbflg=0 : chkflg=0
550   color 6 : locate 20,4 : print "しばらくお待ち下さい。"
560   while stbflg=0
570     v=read(ch)/255#
580     meter(v)
```

```
590     if v<=thrval(2)-0.01# then stbflg=1
600   endwhile
610 /*
620   color 5 : locate 20,4 : print "息を吹き込んで下さい。"
630   while chkflg=0
640     v=read(ch)/255#
650     meter(v)
660     if v>=thrval(2) then v=breath() : check(v) : chkflg=1
670   endwhile
680 endwhile
690 end
1000 /*
1010 /*測定ルーチン
1020 /*（引数）なし
1030 /*（戻り値）測定値
1040 /*
1050 func float breath()
1060   float maxv=0,v=0,v0=0,v1=0
1070   while v1<=v0 and v0<=v
1080     v=read(ch)/255#
1090     meter(v)
1100     if v>=maxv then maxv=v
1110     v1=v0 : v0=v
1120   endwhile
1130   color 5 : locate 28,20 : print using"#.####";maxv*fs
1140   return(maxv)
1150 endfunc
1160 /*
1170 /*酒気チェックルーチン
1180 /*（引数）測定値フルスケール0－1
1190 /*（戻り値）なし
1200 /*
1210 func check(v;float)
1220   if v>=thrval(0) then { busser() } else {
1230       if v>=thrval(1) then {
1240         beep : color 6 : locate 6,4 : print message(1) } else {
1250           color 6 : locate 10,4 : print message(2) }}
1260   color 7 : locate 20,6 : print "何かキーを押して下さい。"
1270   while inkey$=""
1280   endwhile
1290 endfunc
1300 /*
1310 /*警告ルーチン
1320 /*  飲酒運転の警告表示
1330 /*
1340 func busser()
1350   for jjj=1 to 20
1360     beep
1370     color 6  : locate 8,4 : print message(0)
1380     color 14 : locate 8,4 : print message(0)
1390   next
1400   color 6 : locate 8,4 : print message(0)
1410 endfunc
```

リスト3 照度計プログラム

```
10 /* save "d:¥basic¥luxmeter.bas
20 /* save@"d:¥basic¥luxmeter.doc
30 /*
40 /*光センサー応用
50 /*           照度計プログラム
60 /*
70 /*      1991.2.10  K. Misawa
80 /*
90 int cs=0,di=0
100 int xx,yy
110 int rad=300,ox=250,oy=500
120 float rd,elp=256#/360#,fs=5#
130 rd=pi(1)/180#
140 /*
150 float v,vt=0
160 /*
170 /*読み取りチャンネルCH0
180 int ch=0
190 /*
200 screen 1,1,1,1 : window(0,0,511,511)
210 console ,,0
220 vpage(15)
230 ioout(&B0)
240 scale()
250 apage(0)
260 while 1
270   v=lux()
280   meter(v)
290 endwhile
300 end
1000 /*
1010 /*照度測定ルーチン
1020 /*（引数）なし
1030 /*（戻り値）測定値フルスケール0－1
1040 /*
1050 func float lux()
1060   float v,vt=0
1070   for jjj=1 to 10
1080   yy=read(ch) : vt=vt+yy/255#
1090   next
1100   v=vt/10#
1110   return(v)
1120 endfunc
```

リスト4 定時刻照度測定プログラム

```
10 /* save "d:¥basic¥morning.bas
20 /* save@"d:¥basic¥morning.doc
30 /*
40 /*光センサー応用
50 /*  定時刻照度測定プログラム
60 /*
70 /*      1991.2.11  K. Misawa
80 /*
90 int cs=0,di=0
100 int xx,yy
110 int rad=300,ox=250,oy=500
120 float rd,elp=256#/360#,fs=5#
130 rd=pi(1)/180#
140 /*
150 float v,vt=0
160 dim float rec(4,5),frec(5)
170 dim str tm(4,5)
180 /*
190 /*読み取りチャンネルCH0
200 int ch=0
210 /*
220 screen 1,1,1,1 : window(0,0,511,511)
230 console ,,0
240 vpage(15)
250 ioout(&B0)
260 scale()
270 apage(0)
280 /*
290 measure()
300 /*
310 end
500 /*
510 /*定時刻照度測定ルーチン
520 /*  5:00～9:50  10分おき計30回計測
530 /*  データ  "luxval.dat" float型
540 /*          "luxchr.dat" chr型
550 /*
560 func measure()
570   int hh,mm,lv,lc
580   int hhh,mmm,ofs=5
590   lv=fopen("b:luxval.dat","c")
600   lc=fopen("b:luxchr.dat","c")
610   for hhh=ofs to ofs+4
620     for mmm=0 to 5
630     repeat
640       meter(lux())
650       hh=val(left$(time$,2))
660     until hhh=hh
670     repeat
680       meter(lux())
690       mm=val(mid$(time$,4,2))
700     until mm=mmm*10
710     rec(hhh-ofs,mmm)=lux() : frec(mmm)= rec(hhh-ofs,mmm) :
print rec(hhh-ofs,mmm)
720     tm(hhh-ofs,mmm)=time$  : print tm(hhh-ofs,mmm)
730     fwrites(str$(rec(hhh-ofs,mmm))+chr$(13),lc)
740     next
750     fwrite(frec,ofs,lv)
760   next
770   fclose(lc)
780   fclose(lv)
790 endfunc
```

リスト5 おはようアラーム

```
10 /* save "d:¥basic¥alarm.bas
20 /* save@"d:¥basic¥alarm.doc
30 /*
40 /*光センサー応用
50 /*           おはようアラーム
60 /*
70 /*      1991.2.16  K. Misawa
80 /*
90 int cs=0,di=0
100 int xx,yy
110 /*
120 float v,vt=0
130 float setval=0.7#
140 /*
150 /*読み取りチャンネルCH0
160 int ch=0
170 /*
180 screen 1,1,1,1 : window(0,0,511,511)
190 console ,,0
200 vpage(15)
210 ioout(&B0)
220 apage(0)
230 repeat
240   v=lux()
250   print time$;v
260 until v>=setval
270 alarm()
280 end
290 /*
300 /*アラームルーチン
310 /*  好みにしたがって自由に書き替えて下さい
320 /*
330 func alarm()
340   for iii=1 to 100
350     beep
360   next
370 endfunc
```

X68000用 Easy Come, Easy Go!

Yamaoka Kenichi
山岡 賢一

X1/turbo用 シシリエンヌ

Hanai Akitada
花井 章能

あったかくなったり，寒くなったりと，最近は着るものを選ぶのもひと苦労。今月はそんな春一歩手前に似合いそうな曲を2つお届けしましょう。どちらもおそらく聴き覚えがあるはずの曲なので，いままでリストを打ち込んだことのない人も，ぜひ打ち込んで堪能してみてくださいね。

B'zはビーズと読むんだよ

X68000のMUSICDRV用にはB'zの"Easy Come, Easy Go!"をお送りしましょう。この曲はB'zの最新アルバム『RISKY』の中からの選曲で，"銀座じゅわいよ・くちゅ～るマキ"のCMソングにもなっていました。現在このCMは小比類巻かほるさんの曲になってしまっていますが，こちらの曲も聞き覚えがあるのではないでしょうか。ともかく入力して聴いてみることをお勧めします。

B'zは2人組で，その音楽は主に打ち込みなのです。アルバム『RISKY』では打ち込みを感じさせない素晴らしい曲を披露していましたが，山岡君の作品もいい味を出しています。さすがに聴き比べるのはかわいそうな気もしますが，ゲームミュージックとはひと味違ってアレンジのしがいもあったことでしょう。音色の弱さからかオリジナルとはバランスが違うようですが，雰囲気を大切にしてあるのでちゃんと聴けるものに仕上がってます。

'z

さて，山岡君からの指摘なのですが，MUSICDRVはバージョンによって動作が異なる部分があるようです。具体的には，バージョン2.01では@b命令がいままでどおりに使えないこと。PCMのノート番号がひとつずれていること。ほかにもあるそうですが，この2点が大きな問題のようです。これらのせいで，バージョン1.01のミュージックデータはバージョン2.01ではまったく使いものになりません。そこで，今回のプログラムでは簡単なスイッチで両方に対応するようになっています。バージョン2.01を使っている人は180行の"s=0"を"s=1"に変更してください。もしバージョン1.01を使っていて，音色番号のずれを直した人は190行の"n=1"を"n=0"に変更してください。それだけでどちらのバージョンでもちゃんと聴けるようになります。

これは作者からの注文なのですが，バージョン1.01を持っていましたら，そちらのほうで聴いてください。@bコマンドの関係で，より素晴らしく聴くことができるそうです。その際には，BASIC.CNFのMUSIC1.FNCとMUSIC2.FNCの入れ替えも忘れずに行いましょう。

シシリエンヌはパリジェンヌ?

X1用にはガブリエル・フォーレのペリアスとメリザンドより，シシリエンヌをお送りしましょう。この曲はフォーレの中期の小品のひとつで，緩やかなテンポの美しい曲です。ちょっと前にCMにも使われていたと記憶しています。ガブリエル・フォーレを知らない人も多いことと思いますので，軽く触れておきましょう。彼はいまから100年ぐらい前のフランスの人なのです。主な作品としては有名なレクイエムがあげられます。ひょっとしたら心当たりがある人もいるかもしれませんね。あとはラシーヌ讃歌などでしょうか。どちらも宗教音楽ですので，教会音楽と密接に関わっていたといわれるフォーレならではかもしれません。

花井君の作品ではフォーレの慎ましやかで清澄な美しさを見事に伝えるデキになっています。FM音源8声で奏でられるメロディは甘美です。スタッフのある人は電話待ちのオルゴールだなんていってましたけどね。曲は作りやすい構成をしていたのかもしれませんが，非常によくまとまっています。PSGを使っていないのも，曲風を考えると妥当な線といえるでしょう。

この作品のようにもともと音楽性の高い曲を題材にすると音楽に助けられたり，自分がよさを引き出したりと，なかなか楽しくプログラミングできると思います。音楽の解釈は難しいものですが，打ち込みの音楽だからこその解釈のこだわりを聴きたいですね。花井君は1990年7月号のバッハ以来の3回目です。常連を目指してがんばってくださいね。

皆さんへ

投稿の際にはぜひコメントがあるとうれしいなあって思います。作品を聴きながらコメントを読んでいるんですよ。作品名と住所・氏名だけではちょっと寂しいので，苦労した点や，聴いてほしい点，その曲の説明やいままでに載った作品の感想，曲に対する思い入れなどをお寄せください。もちろん乱文乱筆でもOKです。私も字の下手さでは負けません。楽しみに待ってます。

(S.K.)

リスト1 Easy Come, Easy Go!

```
10 /*Save "EasyComeEasyGo!"/*******************************/
20 /*                                                     */
30 /*  E a s y   C o m e , E a s y   G o !       B ' z    */
40 /*                                                     */
50 /*                                                     */
60 /*                 Programmed By  K . Y a m a o k a    */
70 /*                                                     */
80 /*******************************************************/
90 /*
100 m_init() : cls
```

```
110 print "しばらくお待ち下さい。"
120 dim char dm(65000),smp1(4096),smp2(4096),smp3(4096)
130 dim char smp4(4096),smp5(4096),smp6(4096)
140 str m(27)[100],HZ[10] : int p(32),t,n,s : char v(4,10)
150 key 11,"m_tempo(t)@M" : key 20,"m_tempo(300)@M"
160 key 15,"m_play()@M" : key 16,"m_stop()@M"
170 t=118
180 s=0      /* S W 1 MUSICDRV.X ver. 1.0 = 0   2.0 = 1
190 n=1      /* S W 2 ver.1.0 音色番号バグ  無 = 0  有 = 1
200 if s=1 then n=0 : HZ="y49,3" else HZ="@b8170"
210 for i=1 to 9 : m_assign(i,i) : m_alloc(i,4000) : next
220 for i=1 to 8 : m_trk(i,"@b8192p3q8") : next
230 /*
240 /*   FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN
250   v = { 58, 15,  0,  1,200, 15,  0,  4,  0,  3,  0,
260 /*        AR D1R D2R  RR DIL  TL  KS MUL DT1 DT2 AMS
270           25, 14,  3,  3,  1, 28,  2,  3,  5,  0,  0,
280           22, 14,  3,  3, 15, 30,  2,  7,  2,  0,  0,
290           28, 14,  3,  3,  1, 30,  2,  1,  4,  0,  0,
300           18,  6,  3, 10,  2,  0,  1,  1,  6,  0,  1}
310 m_vset(69+n,v)
320 /*   FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN
330   v = { 58, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
340 /*        AR D1R D2R  RR DIL  TL  KS MUL DT1 DT2 AMS
350           26, 11,  2,  2,  5, 38,  1,  3,  7,  0,  0,
360           28, 13,  6,  2,  6, 44,  0, 12,  0,  0,  0,
370           30, 21,  2,  0,  3, 12,  0,  0,  6,  0,  0,
380           26,  6,  0,  6, 15,  0,  2,  1,  3,  0,  0}
390 m_vset(70+n,v)
400 /*   FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN
410   v = {  3, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
420 /*        AR D1R D2R  RR DIL  TL  KS MUL DT1 DT2 AMS
430           31, 13,  3, 10,  8, 39,  0,  3,  6,  0,  0,
440           31, 10,  2, 10,  7,  8,  0,  0,  4,  0,  0,
450           31,  4,  4,  6, 14, 15,  0,  0,  4,  0,  0,
460           28,  3,  5,  8,  3,  0,  0,  1,  3,  0,  1}
470 m_vset(71+n,v)
480 /*    FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN
490   v = {  2, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
500 /*        AR D1R D2R  RR DIL  TL  KS MUL DT1 DT2 AMS
510           30, 16,  0, 10,  0, 30,  0,  2,  0,  3,  0,
520           30, 10,  0, 10, 15, 35,  0,  0,  7,  1,  0,
530           30, 23,  0,  0,  0, 15,  0,  1,  0,  0,  0,
540           30, 18,  5, 10, 13,  0,  0,  1,  0,  0,  0}
550 m_vset(72+n,v)
560 /*   FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN
570   v = { 58, 15,  2,  0,220,  0,  0,  0,  0,  3,  0,
580 /*        AR D1R D2R  RR DIL  TL  KS MUL DT1 DT2 AMS
590           28,  2,  0,  4,  1, 37,  2,  1,  7,  0,  0,
600           22,  4,  1,  1,  1, 43,  2, 13,  0,  0,  0,
610           29,  2,  3,  5,  1, 37,  1,  3,  3,  0,  0,
620           16,  3,  0,  4,  6,  0,  2,  1,  0,  0,  1}
630 m_vset(73+n,v)
640 /*   FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN
650   v = { 28, 15,  2,  0,200,  3,  2,  2,  1,  3,  0,
660 /*        AR D1R D2R  RR DIL  TL  KS MUL DT1 DT2 AMS
670           31,  6,  4,  5,  9, 14,  1,  1,  0,  0,  0,
680           28,  3,  9,  5,  8,  8,  0,  0,  2,  0,  0,
690           31,  4,  7,  5,  6,  4,  0,  0,  0,  0,  0,
700           27,  2,  3,  5,  4,  0,  0,  0,  6,  0,  0}
710 m_vset(74+n,v)
720 /*   FB/AL MSK  WF SYC SPD PMD AMD PMS AMS PAN
730   v = { 58, 15,  2,  0,200, 15,  0,  4,  0,  3,  0,
740 /*        AR D1R D2R  RR DIL  TL  KS MUL DT1 DT2 AMS
750           20, 16,  6,  8,  1, 25,  0,  1,  7,  0,  0,
760           20, 12,  6,  4,  1, 45,  0,  7,  0,  2,  0,
770           22,  0,  6,  4,  0, 43,  0,  1,  0,  0,  0,
780           20,  0,  6, 10,  0,  0,  0,  2,  0,  0,  1}
790 m_vset(75+n,v)
800 /*
810 i=fopen("a:¥smpl¥kick.pcm","r")     /* kick3.6
820  j=fread(smp1,4096,i) : fclose(i)
830   m_pcmset(13-s,4,j,smp1)
840 i=fopen("a:¥smpl¥tom2_5.pcm","r")  /* tom1_4.4
850  j=fread(smp2,4096,i) : fclose(i)
860   m_pcmset(15-s,4,j,smp2)
870 i=fopen("a:¥smpl¥tom2_3.pcm","r")  /* tom1_3.4
880  j=fread(smp3,4096,i) : fclose(i)
890   m_pcmset(17-s,4,j,smp3)
900 i=fopen("a:¥smpl¥tom2_1.pcm","r")  /* tom1_1.4
910  j=fread(smp4,4096,i) : fclose(i)
920   m_pcmset(18-s,4,j,smp4)
930 i=fopen("a:¥smpl¥sn_cr1.6","r")     /* sd_cymv8.pcm
940  j=fread(smp5,4096,i) : fclose(i)
950   m_pcmset(20-s,4,j,smp5)
960 i=fopen("a:¥smpl¥sd5v13.pcm","r")  /* sn2.5
970  j=fread(smp6,4096,i) : fclose(i)
980   m_pcmset(22-s,4,j,smp6)
990 /*
1000 m(0)="@69o5v1318c+e @74v10|:f+e4c+&c+4>b<c+>b<e4c+&c+4>b<c
+>ba4b&b4ab<c+e4c+&c+4c+e
1010 m(1)="f+e4c+4.>b<c+>b<e4c+4.>b<c+>ba4b4.ab<c+e4f+&f+2>@73
1020 m(2)="v13r4c+eee4c+eeeec+16>b16a4.<r4c+e16c+.c+>baba4.r2<
1030 m(3)="r4c+eee4c+eeef+c+16>b16a4.< rc+c+ef+f+&ec+>ba4.r4.a
1040 m(4)="bbbabbbabaaa16f+16&f+4rabbbab.b.<c+e.d.c+>baa4
1050 m(5)="v14bbbabbbabaaa16f+16&f+4r4<eeeef+ec+e4ef+4.rc+e
1060 m(6)="f+eec+4c+>b<c+>b<e4c+4c+>b<c+>ba4b4bab<c+e4c+4.c+e
1070 m(7)="f+16e16eec+4c+>b<c+>bb<ec+rc+>b<c+>bbba16b16&b4ab<c+
eef+4.c+e
1080 m(8)="f+e4c+4.>b<c+>b<e4c+4.>b<c+>ba4b4.abaf+4f+4.v12r<
1090 m(9)="@74<f+4e>babf+ed&d2.g+32&a32&a.g+4.e4e32&f+32&f+.e32
&f+32&f+16&f+1
1100 m(10)="@73v13 rc+c+eee4c+eeee16c+>b16a4.< rc+c+e16c+.c+>ba
ba4.r2<
1110 m(11)="r4c+eee16e.c+eeef+c+16>b16a4.< rc+c+ef+16f+16e4c+>b
a4.r4.a
1120 m(12)="bbbabbbabaaa16f+16&f+4ra16a16bbbab.b.<c+e.d.c+>baa4
1130 m(13)="bbbabb16b16babaaa16f+16&f+4r4<eeeef+ec+e4ef+4.rc+e
1140 m(14)="f+eec+4c+>b<c+>b<e4c+4c+>b<c+>baab4bab<c+c+ec+4.c+e
1150 m(15)="f+16e16eec+4c+>b<c+>bb<ec+rc+>b<c+>ba4b&b4ab<c+eef+
4.c+e
1160 m(16)="f+e4c+4.>b<c+>b<e4c+4.>b<c+>ba4b4.ab<c+e4c+4.c+e
1170 m(17)="f+e4c+4.>b<c+>b<e4c+4.>b<c+>ba4b4.v14ab<c+eef+4.v13
@74<rf+&
1180 m(18)="f+4g+e2c+16e16&e4.e16c+16e32&f+32&f+e16&e4{f+f+f+}2
{f+f+f+}2a4ec+16e16&e4.e l16
1190 m(19)="{c+ec+c+c+}4{c+ec+c+c+}4{c+ec+c+c+}4{c+ec+c+c+}4ec+
f+e{c+c+c+ec+}4c+c>b<c+c>baf+
1200 m(20)="bf+ab<f+c>b32a32f+bf+aba8f+8.r<c+c+f+8f+8&f+2
1210 m(21)="@73v1318o4 r4c+eee4c+eeeec+16>b16a4.< r4c+e4c+>baba
4.r2<
1220 m(22)="r4c+eeeec+eeef+c+16>b16a4.< r4c+ef+e4c+>baba4.r4<
1230 m(23)="v15eeec+e16ee.c+f+ec+f+16&e16&e4.r eeeef+eaaaba16f+
16er4c+e
1240 m(24)="f+16e16eec+4c+>b<c+>bb<ec+4.>b<c+>ba4b4.ab<c+eef+4.
c+e
1250 m(25)="f+e4c+4.>b<c+>b<e4c+4.>b<c+>ba4b4.ab<c+eef+4.c+e
1260 m(26)="f+e4c+4.>b<c+>b<e4c+4.>b<c+>ba4b4.v15ab<c+eef+2&f+
1270 p={32,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,
1280    18,19,20,21,22,23,6,24,6,24,16,25,16,26}
1290 for i=1 to p(0) : m_trk(1,m(p(i))) : next
1300 m(0)=HZ+"@69o5v1518 c+e|:f+e4c+&c+4>b<c+>b<e4c+&c+4>b<c+>b
a4b&b4ab<c+e4c+&c+4c+e
1310 for i=1 to p(0) : m_trk(2,m(p(i))) : next
1320 /*
1330 m(0)="@70o4v1118 r4|:f+4f+f+4f+f+f+d4dd4dddg+4g+g+4g+g+g+e
4ee4eee
1340 m(1)="e4ee4eeed4dd4dddg+4g+g+4g+g+g+f+4f+f+4f+f+f+
1350 m(2)="v9 a4aa4aaaa4aa4aaae4ee4eeee4ee4eee
1360 m(3)="v10 b4bb4bbbf+4f+f+4f+f+f+b4bb4bbbf+4f+f+4f+f+f+
1370 m(4)="b4bb4bbbf+4f+f+4f+f+f+b4.bb4.b4bb2.
1380 m(5)="v11 b4bb4bbbf+4f+f+4f+f+f+g+4g+g+4g+g+g+a4aa4aaa
1390 m(6)="a4aa4aaaf+4f+f+4f+f+f+b4bb4bbba4aa4aaa
1400 m(7)="v12e4.ee4.e4ee2. eeeeeeee v13eeeeeeee
1410 m(8)="a4aa4aaaf+4f+f+4f+f+f+b4bb4bbbv13a4aa4.
1420 p={31,0,1,2,2,3,4,5,5,6,6,2,2,3,4,5,5,
1430    6,6,6,6,2,2,7,5,5,5,5,6,6,6,8}
1440 for i=1 to p(0) : m_trk(3,m(p(i))) : next
1450 /*
1460 m(0)="@70o4v1118 r4|:a4aa4aaa f+4f+f+4f+f+f+ e4ee4eee a4aa
4aaa
1470 m(1)="a4aa4aaa f+4f+f+4f+f+f+ e4ee4eee a4aa4aaa
1480 m(2)="v9 b4bb4bbb b4bb4bbb a4aa4aaa a4aa4aaa
1490 m(3)="v10 e4ee4eee d4dd4ddd e4ee4eee d4dd4ddd
1500 m(4)="@69v8o4 eeec+eeec+eddc+16>b16&b4r4<v10g+g+g+g+ag+g+a
4ab4<r4 v10eg+
1510 m(5)="aaaa4r4. @70o4d4dd4ddde4ee4eeef+4f+f+4f+ @69o5eg+
1520 m(6)="a16a16aaa4r4. @70o4d4dd4ddde4ee4 @69o5rf+ef+g+g+a4.
v12f+g+
1530 m(7)="ag+4f+4.f+g+ag+4f+4.ef+g+f+4e4.f+g+af+4f+4.r4<
1540 m(8)="@70o4v11 f+4f+f+4f+f+f+ d4dd4ddd e4ee4eee @69v10rf+a
brv11<c+ev12f+> @70
1550 m(9)="ag+4f+4.f+g+ag+4f+4.ef+g+f+4e4.f+g+ag+4f+4.f+g+
1560 m(10)="ag+4f+4.f+g+ag+4f+4.ef+g+f+4e4.f+ef+g+g+a4.@70o4v10
r4
1570 m(11)="f+4f+f+4f+f+f+d4dd4ddde4ee4eeef+4f+f+4f+f+f+
1580 m(12)="v12g+4.g+g+4.g+4g+g+2. g+g+g+g+g+g+g+g+ v13g+g+g+g+
g+ @69o5v10reg+
1590 m(13)="ag+4f+4.f+g+ag+4f+4.ef+g+f+4e4.f+ef+g+g+a4.f+g+
1600 m(14)="ag+4f+4.f+g+ag+4f+4.ef+g+f+4e4.v11f+ef+g+g+a2&a
1610 m(15)="a16a16aaa4r4. @70o4 d4dd4ddd e4ee4 @69o5rf+ef+g+g+a
4.eg+
1620 p={31,0,1,2,2,3,4,5,6,7,8,2,2,3,4,5,6,9,10,
1630    11,11,2,2,12,5,15,5,6,9,13,9,14}
1640 for i=1 to p(0) : m_trk(4,m(p(i))) : next
1650 /*
1660 m(0)="@70o4v1118r4|:c+4c+c+4c+c+c+a4aa4aaab4bb4bbbc+4c+c+4
c+c+c+
1670 m(1)="c+4c+c+4c+c+c+a4aa4aaab4bb4bbbc+4c+c+4c+c+c+
1680 m(2)="v9 e4ee4eeee4ee4eeec+4c+c+4c+c+c+c+4c+c+4c+c+c+
1690 m(3)="v10 g+4g+g+4g+g+g+a4aa4aaag+4g+g+4g+g+g+a4aa4aaa
1700 m(4)="g+4g+g+4g+g+g+a4aa4aaag+4.g+g+4.g+4g+g+2.
1710 m(5)="v11 e4ee4eeea4aa4aaab4bb4bbbc+4c+c+4c+c+c+
1720 m(6)="c+4c+c+4c+c+c+a4aa4aaag+4g+g+4g+g+g+c+4c+c+4c+c+c+
1730 m(7)="v12b4.bb4.b4bb2.bbbbbbbbv13bbbbbbbb
1740 m(8)="c+4c+c+4c+c+c+a4aa4aaag+4g+g+4g+g+g+v13c+4c+c+4.
1750 p={31,0,1,2,2,3,4,5,5,6,6,2,2,3,4,5,5,
1760    6,6,6,6,2,2,7,5,5,5,5,6,6,6,8}
1770 for i=1 to p(0) : m_trk(5,m(p(i))) : next
1780 /*
1790 m(0)="@75o5v10l16p1q7 r4|:er8ee8e8ree8e8e8 ar8aa8a8raa8a8a
8 br8bb8b8rbb8b8b8 er8ee8e8
1800 m(1)="ree8eeee er8ee8e8ree8e8e8 ar8aa8a8raa8a8a8 br8bb8b8r
bb8b8b8 er8ee8e8ree8eeee
1810 m(2)="v6q8l8 e.>b.f+4<c+4.&c+1 e.>b.f+4<c+4.&c+1q7
1820 m(3)="v7l16 b8bbb8bbbbbbbbbb a8aaa8aaaaaaaaaa b8bbb8bbbbbb
bbbb a8aaa8aaaaaaaaaa
1830 m(4)="b8bbb8bbbbbbbbbb a8aaa8aaaaaaaaaa b8bbb8bbbbbbbbbb b
8bbb8bbbbbbbbbb v8
1840 m(5)="er8ee8e8ree8e8e8 ar8aa8a8raa8a8a8 br8bb8b8rbb8b8b8 a
r8aa8a8raa8aaaa
1850 m(6)="ar8aa8a8raa8a8a8 ar8aa8a8raa8a8a8 br8bb8b8rbb8b8b8 a
r8aa8a8raa8aaaa
1860 m(7)="v6q8l8 e.>b.f+4<c+4.&c+1 e.>b.f+4<c+4.q7 @74o5v10r4b
64&<c+64&c+32>b16af+.a.c+
1870 m(8)=">a2.r4r1r1< v12rc+a>b<g+>a<f+4 @75
1880 m(9)="@69v9g+g+g+eg+16g+g+.eag+eag+4r4 v11eeeef+eaav12aba1
6f+16e @75r2l16v10
1890 m(10)="ar8aa8a8raa8a8a8 ar8aa8a8raa8a8a8 br8bb8b8rb @74l8o
4v11rab<c+eef+2@75l16o5v10r8
1900 p={31,0,1,2,2,3,4,5,5,6,6,7,8,3,4,5,5,
1910    6,6,6,6,2,2,9,5,5,5,5,6,10,6,10}
```

▶ラプラスの魔のレビューのところ，名前が載っていないけど，たぶん古村さんでしょう。
工藤　秀行(23)東京都

```
1920 for i=1 to p(0) : m_trk(6,m(p(i))) : next
1930 /*
1940 m(0)="@71o3v1318 r4 f+f+f+f+f+f+ee dddddf+f+ eeeeeeg+g+ f
+f+f+f+f+f+c+c+
1950 m(1)="f+f+f+f+f+f+eedddddf+f+eeeeeeg+g+f+f+f+f+f+f+<g+4>
1960 m(2)="v11 a4ra&a2 a4ra&a2 f+4rf+&f+2 f+4rf+&f+2
1970 m(3)="14v12 eeee dddd eeee dddd
1980 m(4)="eeee dddd er8e8er8ee8e.v13<e>e8
1990 m(5)="18 aaaaaac+c+ddddddf+f+eeeeeeg+g+f+f+f+f+f+f+g+g+
2000 m(6)="aaaaaac+c+ ddddddf+f+ eeeeeeg+g+ f+f+f+f+f+f+ee
2010 m(7)="f+f+f+f+f+f+eedddddf+f+eeeeeeg+g+f+f+f+f+f+f+c+c+
2020 m(8)="f+f+f+f+f+f+eedddddf+f+eeeeeeg+g+f+f+f+f+f+f+g+g+
2030 m(9)="f+f+f+f+f+f+eedddddf+f+eeeeeeg+g+f+f+f+f+f+f+ee
2040 m(10)="f+f+f+f+f+f+eedddddf+f+eeeeeeg+g+rf+f+f+f+g+g+
2050 m(11)="v10<al&al f+l&f+1
2060 m(12)="al&al f+1&f+4 v13g+2.>
2070 m(13)="e4ree4re4ee4.<e4>v12eeeeeeee v14eeeeeeeeev13
2080 m(14)="f+f+f+f+f+f+eedddddf+f+eeeee<ev15g+>g+f+f+4f+2&f+
2090 p={31,0,1,2,2,3,4,5,6,7,8,2,2,3,4,5,6,7,9,7,
2100    10,11,12,13,5,5,5,6,7,7,7,14}
2110 for i=1 to p(0) : m_trk(7,m(p(i))) : next
2120 /*
2130 m(0)="@72v15o5rl16p2|:4aaaaa8aaa8aaa8aaaaaaa8aaa8aaaaaa:|
2140 m(1)="|:14a8aaa8aaa8aaa8aa:|r1r1
2150 m(2)="|:8aaaaa8aaa8aaa8aaaaaaa8aaa8aaaaaa:|
2160 m(3)="|:4aaaaa8aaa8aaa8aaaaaaa8aaa8aaaaaa:|
2170 m(4)="v13|:8a8aaa8aaa8aaa8aa:|
2180 m(5)="r1r1r1r1 v15|:15aaaaa8aaa8aaa8aaaaaaa8aaa8aaaaaa:|
2190 m(6)="aaaaa8aaa8aaaaaa a8aaa8a8
2200 p={9,0,1,2,1,2,3,4,5,6}
2210 for i=1 to p(0) : m_trk(8,m(p(i))) : next
2220 /*
2230 m(0)="o014 g ga8c8ca ca8c8ca ca8c8ca cac8e8d
2240 m(1)="ga8c8ca ca8c8ca ca8c8ca cac8e16e16a
2250 m(2)="ga8c8ca ca8c8ca ca8c8ca ca8c8ca
2260 m(3)="ca8c8ca ca8c8ca ca8c8ca ca8c8c8.a16a8c8
2270 m(4)="caca caca caca cac8d8e
2280 m(5)="caca ccc8.f16e8d8 ga8g8ga8g8 r8g8g8f8e8d16d16d8c8
2290 m(6)="ga8c8ca caca caca cac8c8a
2300 m(7)="ga8c8ca ca8c8ca ca8c8ca cac8a16a16a16a16a8
2310 m(8)="ga8c8ca ca8c8ca ca8c8ca cac8c8a8a16a16
2320 m(9)="ga8c8ca ca8c8ca ca8c8ca cac8c16a16a
2330 m(10)="ca8c8ca ca8c8ca ca8c8ca ca8c8c8a8a8a16a16
2340 m(11)="caca caca caca cac8e8a8d8
2350 m(12)="caca ca8a16a16f16f16e16e16d8c8 ga8g8ga8g8 r8g8g8f16
f16e8d8d8c8
2360 m(13)="ga8c8ca ca8c8ca ca8c8ca cac8c8a
2370 m(14)="ga8c8ca ca8c8ca ca8c8cr8c16c16 18accgaal16aaaal4
2380 m(15)="ga8c8ca ca8c8ca ca8c8ca cac8c8a8c8
2390 m(16)="ga8c8ca ca8c8ca ca8c8ca ca8l16aaddddeee814
2400 m(17)="r1r1r1r1
2410 m(18)="r1r1r1rr8.f16f8e8d8c8
2420 m(19)="ga8g8ga8g8r8g8g8a8f8.e16e4a8a8a8a8a8a8a8a8a8a8a8a
16a16c16c16g8c8
2430 m(20)="ga8c8ca ca8c8ca ca8c8ca l16ccf8f8ccf8eeddd814
2440 m(21)="ga8c8ca ca8c8ca ca8c8ca caca8g8
2450 m(22)="ga8c8ca ca8c8ca ca8c8ca cac8c8g
2460 m(23)="ga8c8ca ca8c8ca ca8c8ce8d8 g8f16f16d8g4.
2470 p={31,0,1,2,3,4,5,6,7,8,9,2,10,11,12,6,7,13,
2480    14,15,16,17,18,19,6,7,6,20,21,15,22,23}
2490 for i=1 to p(0) : m_trk(9,m(p(i))) : next
2500 /*
2510 m_tempo(t)
2520 m_play()
2530 /*m_mute(1,2,3,5,6,7,9)
2540 /*
2550 print "［Ｆ１５］でＰＬＡＹ、［Ｆ１６］でＳＴＯＰします。
```

リスト2 シシリエンヌ

```
10 REM ********************************
20 REM *                              *
30 REM *          SICILIENNE          *
40 REM *  de "Pelleas et Melisande"   *
50 REM *                              *
60 REM *              Gabriel Faure   *
70 REM *                              *
80 REM ********************************
90 TEMPO 0
100 DEFSTR A-I:DEFINT J-Z
110 MEM$(&HB190,36)=HEXCHR$("F900060472012F343400549454511E0A058
801010101F5F5F5C900000080F4C880000200")
120 MEM$(&HB1B4,36)=HEXCHR$("F900060472012F3439005494544E1E0A058
801010101F5F5F5C800000080F4C880000200")
130 GOTO 260
140 '
150 LABEL "!"
160 ON J GOTO 170,180,190,200,210,220,230,240
170 PLAY A;:RETURN
180 PLAY B;:RETURN
190 PLAY C;:RETURN
200 PLAY D;:RETURN
210 PLAY E;:RETURN
220 PLAY F;:RETURN
230 PLAY G;:RETURN
240 PLAY H;:RETURN
250 '
260 FOR J=1 TO 8
270 A1="O2G>G>D<FB->G<E>DB-<F>DA"
280 A="T72 I1Q8P3K5 V9L8O2G>G>D<<G>G>D"+A1
290 B1="R16O3DB-8.R16G>D8.R16<B->G8.R16<A>F8."
300 B="I1Q8P3K5 V9L8O3R16DB-8.R16DB-8."+B1
310 C1="O2G>DGB->D8<FGB->DG8<EB->DGB-8<FA>DFA8"
320 C="I2Q8P1K12 V7L16O2R16."+STRING$(2,"G>DGB->D8<<")+C1
330 D1="O4DG4B->D4GB-8.A16GA4"
340 D="I1Q8P3K5 V11L8R2R"+D1
350 E="I2Q8P2K12 V9L8R2RR16."+D1
360 F="R1R1R4"
370 G=F:H=F
380 "!"
390 A2="O2B-4.B-4.D>DA>F+8R4"
400 A=A2
410 B2="O3R16F16B-GR16F16B-GR16<A>F+>DR16R4"
420 B=B2
430 C2="O2B->FB-8G8<B->FB-8G8<DA>DF+A>DF+8R4"
440 C=C2
450 D2="D&D8.C16ED8.C16ED2R"
460 D=D2:E=D
470 F="I1Q8P3K5 V9L8O4RFERFEF+RR2"
480 G="I1Q8P3K5 V9L8O4RDCRDCDRR2"
490 H="I1Q8P3K5 V9L8O3RFGRFGDRR2"
500 "!"
510 A3="O2G>G>D<FG>D<E->CA-<DB>G"
520 A=A3
530 B3="O3R16DB-8.R16>D<B-8.R16A->F8.R16<G>F8."
540 B=B3
550 C3="O2G>DGB->D8<F>D<GB->D8<E-A->CFA-8<DGB>FG8"
560 C=C3
570 D3="O4DG4¯2B-¯1>D4¯2F¯1A-8._4G16F_2G4"
580 D=D3:E=D
590 F="R2.A-4.G4"
600 G="I2P2K12 V7R16.R2.A-4.G4"
610 H="R1R2"
620 "!"
630 A="O3C4.D4.T70<G>B-B-<T68G>B->DT72"
640 B="R16O3G16B-4R16A16>C4R16<DG>D<DG8."
650 C="O3CGB-4DA>C4<<G>DB-GB->D<<G>DB-G>D8"
660 D4="<G&G8.F+16AG8.F+16B-G2R"
670 D="V9"+D4
680 E="V7"+D4
690 F="FE4E-D4F+G2R4"
700 G=F
710 H="O3RB->E-RCDDRR2"
720 "!"
730 A=A1:B=B1:C=C1
740 D="V11"+D1:E="V9"+D1
750 F="G4.F4.E4.F4R"
760 G=F+"32"
770 H="R1R2"
780 "!"
790 A=A2:B=B2:C=C2
800 D=D2:E=D
810 F="RFERFEF+RR2"
820 G="I1P3K5 V9RDCRDCDRR2"
830 H="O3RFGRFGDRR4>DC"
840 "!"
850 A=A3:B=B3:C=C3
860 D=D3:E=D
870 F="O3B-RR2>A-4.G4"
880 G="I2P2K12 V7O4R16.R2.A-4.G4"
890 H="O3G4.F4.E-4.D4R"
900 "!"
910 A="O3C4.D4.T70<G>B-B-T68<G>BBT72"
920 B="R16O3G16B-4R16A16>C4R16<DG>D<DG>D16"
930 C="O3CGB-4DA>C4<<G>DB-GB->D<<G>DBGB32"
940 D="V10"+D4+"4":E="V8"+D4+"R32"
950 F="_4>FE4E-D4<F+G2RR"
960 G=F+"32"
970 H="RB->E-RCDDRR2"
980 "!"
990 A5="O4B8.G16AB-8.>C16C+D8.E-16FG4D<"
1000 A="I1P3K5 V12"+A5
1010 B="I2P1K12 V10R16."+A5
1020 C="I1P3K5 V12L8O4G2.F4E-D4.&"
1030 D="I2P1K12 V10O4R16.G2.F4E-D4.&"
1040 I="O4D4D-C4<B-A4.B-8.A16A-"
1050 E="I1P3K5 V12"+I
1060 F="I2P1K12 V10R16."+I
1070 G="I1P3K5 V12L4.O3EE-F<B->"
1080 H="I2P1K12 V10L4.O3R16.EE-F<B->"
1090 "!"
1100 A6="B-8.G16AB-8.>C16C+D8.E16F+G4"
1110 A=A6+".":B=A
1120 C="D4.G4.F+8.G16AG4."
1130 D=C
1140 E6="G4>DC4<B-A8.B-16>C<B-4"
1150 E=E6+".":F=E
1160 G="EE-D<G>"
1170 H=G
1180 "!"
1190 A="V10"+A5:B="V8"+A5
1200 C="V10G2.F4E-D4."
1210 D="V8G2.F4E-D4."
1220 I="O4D4D-C4<B-A4.A-8.G16A-"
1230 E="V10"+I:F="V8"+I
1240 G="V10EE-F<B->"
1250 H="V8EE-F<B->"
1260 "!"
1270 A=A6+"T68RT72":B=A
1280 C="G2.F+8.G16AG4R"
1290 D=C
1300 E=E6+"R":F=E
1310 G="EE-D<G4R8"
1320 H=G
1330 "!"
1340 A8="O5G4._7D4RB-8.A16B-A8.G16F+"
```

▶入試前から，某予備校から「君はローニンしても特待生として優遇しちゃうよーん」という通知が来て以来，妙に安心している私であった（だって，１年分の授業料をタダにしてくれるっていうんだもの……）。
田端　秀章(18)北海道

```
1350 A="V13"+A8:B="V10"+A8
1360 C8="O5E-4._7<B-4R>G8.F+16GF+8.E16D"
1370 C="V13"+C8:D="V10"+C8
1380 E8="O3A4._7B-4R>>C+4.<A4."
1390 E="V13"+E8:F="V10"+E8
1400 G8="O2F4._7G4R8L16A>EA>C+E8<DA>DF+A8"
1410 G="V13"+G8:H="V10"+G8
1420 "!"
1430 A9="G4._7D4RC8.<A16B-AL16¯1D¯F+¯A¯>DL8"
1440 A="V13"+A9:B="V10"+A9
1450 C9="E-4._7<B-4RA8.F+16GF+R4"
1460 C="V13"+C9:D="V10"+C9
1470 E9="O3A4._7B-4R>C4C+<AR4"
1480 E="V13"+E9:F="V10"+E9
1490 G9="O2F4._7G4R8A>EGB->C+EF+4."
1500 G="V13"+G9:H="V10"+G9
1510 "!"
1520 A="V13"+A8:B="V10"+A8
1530 C="V13"+C8:D="V10"+C8
1540 E="V13"+E8:F="V10"+E8
1550 G="V13"+G8:H="V10"+G8
1560 "!"
1570 A="G4.D4RT70C8.<A-16B->T65C4D"
1580 B=A+"32"
1590 C="E-4.<B-4RA-4GF+8.E16F+"
1600 D="E-4.<B-4RA-4GF+8&F+32"
1610 E="O3A4.B-4R>D4C<B-4"
1620 F=E+">F+"
1630 G="O2F4.G4R8>F4E-8D4D8"
1640 H=G
1650 "!"
1660 A="I1P3K5 V9"+A1
1670 B="I1P3K5 V9"+B1
1680 C="I2P1K12 R16.V7L16"+C1
1690 D="T72 I1P3K5 V12"+D1
1700 E="I2P2K12 R16.V10"+D1
1710 F="R32R4.R1"
1720 G="R1R2"
1730 H=F
1740 "!"
1750 A=A2:B=B2:C=C2
1760 D="V11"+D2:E="V9"+D2
1770 F="I1P3K5 V9L8O4RFERFEF+RO3AB-4>C"
1780 G="I1P3K5 V9L8O4RDCRDCDRR2"
1790 H="I1P3K5 V9L8O2B->FG<B->FGDR I2P1K12R16.O3AB-4>C"
1800 "!"
1810 A=A3:B=B3:C=C3
1820 D=D3:E=D
1830 F="<B-2R2>C<B4>F"
1840 G="R4O4DG4B->C4<A-G4R"
1850 H="<B-4>DG4B->C4<C<B4>F"
1860 "!"
1870 A="O3C4.D4.T70<G>B-B-O1T68B->>DT65B-T72"
1880 B="R16O3G16B-4R16A16>C4R16<DG>D<<B->G>D16"
1890 C="O3CGB-4DA>C4<<G>DB-GB->DO1B->B->DGB-32"
1900 D="V10<G&G8.F+16AG8.F+16B-G2."
1910 E="V9<G&G8.F+16AG8.F+16B-G2&G8&G32"
1920 F="E4E-D4F+G2."
1930 G="<RB->E-RCD<GRR2"
1940 H="E4E-D4F+G2&G8&G32"
1950 "!"
1960 A="I1P3K5 V10O4G2.G2."
1970 B="I2P3K12 R16.V7O4G2.G2."
1980 C="I1P3K5 V10O4E-2.E-2."
1990 D="I2P3K12 R16.V7O4E-2.E-2."
2000 E="I1P3K5 V10O2B-2.B-2."
2010 F="I2P3K12 R16.V7O2B-2.B-2."
2020 G="I1P3K5 V10O3G2.G2."
2030 H="I1P3K5 V10O2E-2.E-2."
2040 "!"
2050 I="O6E-8.D16C<B-8.A-16GF8.G16EF8.D-16DE-8.F16DE-8.<B16>D-E-
4.<B-R4"
2060 A="V11"+I+"F2.":B="P1 V9"+I+"P2F2."
2070 I="O2E-4.O4E-4RG2.G4.F4.G4.G4A-B-2."
2080 C="V11L8"+I:D="P2 V9"+I
2090 I="O3G4A-B-4>CD-4CD-4<B->C4<E-B4A-B-4G8E-4<E-B-2."
2100 E="V11"+I:F="P2 V9"+I
2110 I="O1E-4.O3G4E-E-2.E-2.E-4.G4RA-4.F4G"
2120 G="V11"+I:H="I2P2K12 R16.V9"+I
2130 "!"
2140 A="O6F8.E-16DC8.<B-16A-G8.D16EF+8.G16B-F4.<B-4."
2150 B="P1"+A
2160 C="O3A-4.B-4>CD2&DGF2."
2170 D=C
2180 E="O4D4.R4.<G4.B-4.A-4<B->F4G"
2190 F=E
2200 G="O2B-2.B-4.>G4.<B-4R2"
2210 H=G
2220 "!"
2230 A="O6F8.E-16DC8.<B-16A-G8.E16F+G8.A-16B->E-4.<E-4.&"
2240 B=A
2250 C="O3A-4.B-4>CB-4AG4F<G4<B->E-4F"
2260 D=C
2270 E="O4D4.R4.D4.D4DO2B-4R2"
2280 F=E
2290 G="B-2.>B-4.B-4A-<E-4R2"
2300 H=G
2310 "!"
2320 A="E-8.>D16C<B-8.A-16GF8.G16EF8.D-16DE-8.F16D¯2E-8.<¯B16>E-
"
2330 B=A
2340 C="G4A-B-4>CD-4CD-4<B->C4<E-¯2B4¯B-"
2350 D=C
2360 E="O4G2.G2.G4.¯2F4.¯"
2370 F=E
2380 G="O2B-4R2>E-2.E-2.¯4"
2390 H=G
2400 "!"
2410 A=STRING$(2,"G4._4<A8.B16>D-E-8.D-16E-¯2F8.¯E-16F")
2420 B=A
2430 C="O4G4._4G-2.¯2B-4.¯G4._4G-2.¯2B-4."
2440 D=C
2450 E="E-2.&E-4._2G4.¯2E-2.&E-4._2G4."
2460 F=E
2470 G="A8.G16A_4B2.¯2B-4.¯A8.G16A_4B2.¯2B-4."
2480 H=G
2490 "!"
2500 A="V13"+A8:B="V10P2"+A8
2510 C="V13"+C8:D="V10"+C8
2520 E="V13"+E8:F="V10"+E8
2530 G="V13"+G8:H="V10"+G8
2540 "!"
2550 A="V13"+A9:B="V10"+A9
2560 C="V13"+C9:D="V10"+C9
2570 E="V13"+E9:F="V10"+E9
2580 G="V13"+G9:H="V10"+G9
2590 "!"
2600 A="V13"+A8:B="V10"+A8
2610 C="V13"+C8:D="V10"+C8
2620 E="V13"+E8:F="V10"+E8
2630 G="V13"+G8:H="V10"+G8
2640 "!"
2650 A="G4.D4RT70C8.<A-16B->T65C4D"
2660 B=A+"32"
2670 C="E-4.<B-4RA-4GF+8.E16F+"
2680 D="E-4.<B-4RA-4GF+8&F+32"
2690 E="O3A4.B-4R>D4C<B-4"
2700 F=E+">F+"
2710 G="O2F4.G4R8>F4E-8D4D8"
2720 H=G
2730 "!"
2740 A="I1P3K5 V9"+A1
2750 B="I1P3K5 V9"+B1
2760 C="I2P1K12 R16.V7L16"+C1
2770 D="T72 I1P3K5 V12"+D1
2780 E="I2P2K12 R16.V10"+D1
2790 F="R32R4.R1"
2800 G="R1R2"
2810 H=F
2820 "!"
2830 A=A2:B=B2:C=C2
2840 D="V11"+D2:E="V9"+D2
2850 F="I1P3K5 V9L8O4RFERFEF+4R2"
2860 G="I1P3K5 V9L8O4RDCRDCD4R2"
2870 H="I1P3K5 V9L8O3RFGRFGD4R4>DC"
2880 "!"
2890 A=A3:B=B3:C=C3
2900 D=D3:E=D
2910 F="O3RB->D4GB->CR<A-G4F"
2920 G="I2P1K12 R16.V7"+F
2930 H="G4.F4.E-4.D4."
2940 "!"
2950 A="O3C4.D4.E-B->G<D-B->G"
2960 B="T70R16O3G16B-4T68R16A16>C4T72R16<G>E-8.<R16G>E-8."
2970 C="O3CGB-4DA>C4<E-GB->E-G8<D-GB->E-G8"
2980 I="<G&G8.F+16AG8.F+16B-G2&G"
2990 D="V9"+I:E="V7"+I
3000 F="E4E-D4F+G2."
3010 G=F
3020 H="R<B->E-RCD<L4.E-D-"
3030 "!"
3040 A="O2B->G>E-<<A>E-R<G>E-R<F>E-R<E->E-RD4R"
3050 B="O3R16E-B-8.R16<B->B-R<B->B-R<B->B-R<B->B-RA16>CR"
3060 C="O2B->E-GB->E-8<<AB->E-B-R8<GB->E-B-R8<FB->E-B-R8<E-B->E-
B-R8DA>C8R8"
3070 I="O3GB-4>E-G4B->E-4FG4AB-4.&B-8._5F+16AG4"
3080 D="V11"+I:E="V9"+I
3090 F="R4.G4.&G4A-B-4.&B-4>CD8.<A16>C<B-4"
3100 G=F
3110 H="<B-AGFE-R"
3120 "!"
3130 A="O2G>B-<A-G>B-<FE->B-<E-D>F+R"
3140 B="O2R16B-B-8.R16B-B-8.R16B-B-8.R16>D8.R"
3150 C="O2GB->B-<B-A-8GB->B-<B-F8E-B->B-<B-E-8D>DF+8R8"
3160 I="O4RG4A-B-4>CD8.<F+16A"
3170 D="V9"+I:E="P1V7"+I
3180 F="O3B->E-4FG4AB-8.<A16>C"
3190 G=F
3200 H="L8O2G4A-G4FE-4RD4R"
3210 "!"
3220 A="O2"+STRING$(4,"G>GC<")+"T68G4."
3230 B="O3"+STRING$(4,"R16D16GC")+"<G4."
3240 C="O2"+STRING$(4,"G>DG8C8<")+"G4."
3250 D=STRING$(2,"G4E-D8.F16E")+"D4."
3260 E=D
3270 F="O3B-8.>C16<AB-4.>D8.C16<AB-4.B-4."
3280 G=F
3290 H=STRING$(4,"G4>C<")+"I2P1K12 R16.V7G4."
3300 "!"
3310 A="R16.V8D4.D2."
3320 B="T60V8D4.D2.&D16."
3330 C="R16.V6D4.D2."
3340 D="R16V8<B-4.B-2.&B-32"
3350 E="R16V6<B-4.B-2.&B-32"
3360 F="R32V8G4.G2.&G16"
3370 G="R32V6G4.G2.&G16"
3380 H="V6D4.D2.&D16."
3390 "!"
3400 PLAY ":";
3410 NEXT
3420 PLAY ""
```

育てや育て，プログラム

Komura Satoshi 古村 聡

illustration : T.Takahashi

今回のショートプロは先月少し紹介した迷路自動作成プログラムがまず１本目。そして，特別にスタッフの福原君からのショートデモプログラムが２本目ということです。また，ぱーてぃハンズの3D迷路がいよいよ完成します。

どもども，(で) です。

実は，ここだけの話なんですけど。時代遅れにもいまさら「シムシティー」に凝っている私だったりします。

「シムシティー」とか，今度X68000にも出るはずの「A列車で行こうIII」とか，いいですね。私のようなドジでマヌケな人にも育てる喜びが簡単に味わえますもん。失敗してもリトライできるし，人の作ったもんだから繰り返しやってればなんとかある程度まで育つもんだし。

しかし，本物の生き物のほうはといえば，いつかはそうなるものだとはいえ，飼っていたものが死んじゃったりするとすっごくこたえるもんなんですよね (私も動物を不注意で死なせたりしたものです)。それで生き物を飼育するのが苦手になっちゃった子供っていうのもどこかにいるんじゃないかなんて思います。このテのゲームってもしかしたらそういう子供たちの自信回復にも使えるんじゃないでしょうか？　僕にもちゃんとここまで街を育てることができたんだとかってね。

……ああ。にしてもなんだなあ。もう１回，本当の生き物を飼ってみたい気もするなあ。今度は金魚でも，死なないようにちゃんと世話するもんね。

お待たせいたしました！

１カ月のお待たせでした。１本目は予告していたとおり，アセンブラによる迷路自動作成プログラムです。

rmaze3.x for X68000

(要アセンブラ)

東京都 阿妻 靖史

このプログラムではX68000付属のエディタ，「ed.x」のほかにアセンブラ「as.x」，リンカ「lk.x」，マクロデータ「DOSCALL.MAC」，「IOCSCALL.MAC」が必要です。「as.x」，「lk.x」は「C compiler PRO-68K」，あるいは「福袋 ver.2.0」についてきます。「DOSCALL.MAC」と「IOCSCALL.MAC」はC compiler PRO-68K，あるいは本誌の付録ディスク「創刊８周年記念 PRO-68K」，「謹賀新年 PRO-68K」などに収録されています。

使い方ですが，いちばん簡単なのは，ディスク上に「as.x」，「lk.x」，「ed.x」，「DOSCALL.MAC」，「IOCSCALL.MAC」を同じディレクトリ上に置きます。そして，エディタから「rmaze3.s」というファイル名でリストを打ち込み，

```
A>as rmaze3
A>lk rmaze3
```

とすれば実行ファイルを作ることができます (アセンブル，リンクのときにエラーが出たら，エディタに戻って間違えているところを探して打ち直す)。

```
A>rmaze3
```

として実行すると……，お，おおお？

画面写真を見ればわかりますが，ソバ状のものがにゅにゅにゅっと伸びてきて画面いっぱいに１ドットの太さのラインで迷路を作ってくれるのですね。すごい。じっとながめてると脳味噌かなんかのようだ。

投稿原稿によると，このプログラムは再帰呼び出しを使ったアルゴリズムによって迷路を描き出しているのだそうです。ふむふむ，rmazeの“r”はrecursive (再帰的) のｒなのね。アルゴリズムとしては，

1)　スタート

2)　自分のいる座標の周りを調べ，四方がすでに通路だったらリターン

3)　進むことのできる方向からランダムに自分の進む方向を選ぶ

4)　現在の位置をスタックして１)，つまり自分自身にサブルーチンコールする

5)　自分の呼んだサブルーチンが掘っていた場所が通路で囲まれているかもしれないので，もう一度自分のいる座標の周りを調べ (ついでにスタックがあふれていないか調べて)，四方が通路だったらリターン

6)　そうでなかったら２)に戻る

となっているんですね。

再帰呼び出しっていうのは，サブルーチンの中でもう一度そのサブルーチン (つまり自分自身) を呼び出す方法をいうのです。これはスタックに自由にワークエリアを確保できる68000のマシン語や，サブルーチンの中でローカル変数を定義できるＣやPascalの得意技だったりするのですよね。ちなみにこのプログラムだと自分で自分を軽く２万回以上呼んでいるんだそうです。

関係ないけど，これってパーティハンズの3D迷路表示ルーチンと組み合わせたらどうなるんでしょうね？　ものすごくというか，すさまじい広さの迷路ですからね。迷って，一生出られないかもしれない。

にしても阿妻さん，先月もいったような気もするけど，この迷路作成シリーズだけで６本もマシン語プログラムを送ってきてくださってんですよね。(で)はただただ感謝するばかりであります。ありがとうございました。うるうる。

スタッフからショートプロ

迷子になってしまう前に次いきましょう。今月の２本目はOh!Xスタッフ，福原徹氏のX68000用デモプログラム，wak.cです。

wak.c for X68000

(要Cコンパイラ)

埼玉県　福原　徹

なんと，いきなりスタッフの作品という反則技だったりします。ショートプロはルールのない格闘技だ！　……なんてね。

ま，それはともかく，この作品なのですが，なんと宇宙を育てるという壮大なテーマだったりするのです。

打ち込み方ですが，このプログラムは市販の「C compiler PRO-68K ver.2.0」かGCC(ただし，こちらの場合もXCのライブラリが必要)が必要になります。とりあえずここではXCでのコンパイルについてふれておきます。

コマンドラインのほうが楽ですので「command.x」が使える状態にしてください。

A>

こういう表示が出ていますね。次にエディタを立ち上げます。「ed.x」というのがそれですので「ed　ファイル名」と打ち込みます。

A>ed wak.c

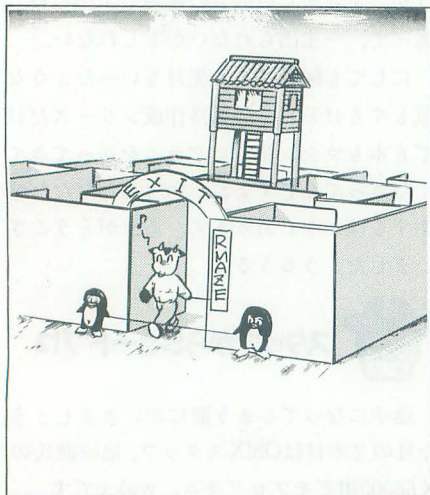
で，打ち終わったらESC+Eでエディタを抜けてコンパイルします。

A>cc /Y /W wak.c

と打ち込んでください。エラーが出なかったら，実行です。

(gccの場合は -traditional -fstrength -reduce -finline -functions -fomit -frame -pointer -lbas -liocs -ldos のオプションをつけてコンパイルしてください)

A>wak

さあ，なにが見えるかな？

リスト1　rmaze3.s

```
  1:         .include DOSCALL.MAC
  2:         .include IOCSCALL.MAC
  3:
  4: PEN_COLOR       equu    %11000_01111_01111_1
  5: PAPER_COLOR     equ     %00011_00011_00011_0
  6: __RAND          equ     $FE0E
  7: pen             equ     1
  8: paper           equ     0
  9: GRAMADR         equ     $C00000
 10: V_1DOT          equ     $800
 11:
 12:         .text
 13:         .even
 14:
 15: enter:
 16:         lea     bigstack,sp
 17:
 18: init:                                                   *画面関係の初期化
 19:         moveq.l #16,d1
 20:         IOCS    _CRTMOD
 21:
 22:         IOCS    _G_CLR_ON
 23:
 24:         moveq.l #paper,d1
 25:         move.l  #PAPER_COLOR,d2
 26:         IOCS    _GPALET
 27:
 28:         moveq.l #pen,d1
 29:         move.l  #PEN_COLOR,d2
 30:         IOCS    _GPALET
 31:
 32:         moveq.l #%0001,d1
 33:         IOCS    _VPAGE
 34:
 35:         moveq.l #0,d1
 36:         IOCS    _APAGE
 37:
 38:         lea     f_param,a1
 39:         IOCS    _FILL
 40:
 41:         lea     f_param2,a1
 42:         IOCS    _FILL
 43:
 44:         clr.l   a1                                      *スーパーバイザモードにして
 45:         IOCS    _B_SUPER
 46:         move.l  d0,userstack
 47:
 48: main:
 49:         move.l  #V_1DOT,d3                              *d3:Y 1dot displacement
 50:         lea     flowchk,a3                              *a3:stack overflow check
 51:         lea     GRAMADR+V_1DOT*254+2*382,a4             *a4:画面の真ん中のアドレス
 52:         move.w  #pen,d5                                 *d5:ペン（通路の色）
 53:         moveq.l #0,d7                                   *d7:乱数用のループカウンタ
 54:         move.w  d5,(a4)                                 *はじめのいっぽ
 55:
 56:         pea     (a4)                                    *今いる位置を渡して
 57:         bsr     makemaze                                *迷路を描くルーチンを呼び出す
 58:         addq.l  #4,sp                                   *スタック補正
 59:                                                         *最初これを忘れてえらい目にあった
 60:
 61: exit:
 62:         movea.l userstack,a1                            *リターンしてきたらすぐに終了
 63:         IOCS    _B_SUPER
 64:
 65:         DOS     _EXIT
 66:
 67: makemaze:
 68:         movea.l 4(sp),a4                                *A4を取り出す（この行は消しても
 69:                                                         *動くような気がする）
 70:         bsr     rand                                    *実際に描かせる部分
 71:
 72:         bsr     chkrot                                  *周りを調べる
 73:         tst.w   d0                                      *D0=0なら四方塞がり
 74:         beq     return
 75:
 76:         cmpa.l  a3,sp                                   *スタック溢れのチェック
 77:         bcs     return
 78:
 79:         pea     (a4)                                    *さらに下位のサブルーチン（自分）
 80:         bsr     makemaze                                *を呼ぶ
 81:         addq.l  #4,sp                                   *再びスタック補正（重要）
 82:
 83:         movea.l 4(sp),a4                                *A4は壊されるので（高速化とスタッ
 84:         bsr     chkrot                                  *クの節約の為）もう一度取ってくる
 85:         tst.w   d0                                      *まだ進めるなら
 86:         bne     makemaze                                *もう一度戻って迷路を描く
 87:
 88:
 89: return:
 90:         rts                                             *レジスタの保存もスタックフレームも
 91:                                                         *ないのでこれだけで良い
 92: chkrot:
 93:         moveq.l #1,d0                                   *まだ壁がどこかにあれば1を返す
 94:
 95:         cmp.w   4(a4),d5                                *2ドット右を調べる
 96:         bne     pass
 97:
 98:         cmp.w   V_1DOT*2(a4),d5                         *下を調べる
 99:         bne     pass
100:
```

きれいな幾何学模様が出てきたでしょう。
ESCキーを押すことでこのプログラムから抜けることができます。

さて，投稿原稿（そう，たとえスタッフの作品といえども原稿は必要なのです）によれば……，

“てなわけで，NHKの宇宙教育番組のオープニングCGを見ていて，銀河内の星の重力シミュレーションプログラムをX68000でやってみようかなあなんて，大それたこと思ったのがこのプログラムができたきっかけなわけです，実は（普通ならこういうのってスパコンクラスのコンピュータを使うんだよな）。

「処理速度が遅いよお，重いよお」。だが，開発中になにやらきれいなデモができてしまったではないか！　リストも短めだし，「こりゃぁショートプロぱーてぃに投稿するっきゃないね」ってなわけで投稿したっつうわけです。結局本当の目的であるシミュレーションは挫折したままであったりする(笑)。”

ほっほっほっ，いいじゃないですか。そういう大風呂敷，私は大好きです。

それにしても奇っ怪な構造してますね。リスト見るとわかるんだけど構造体の中にポインタがあるんだもんね。

なんでも，

“星のデータを格納している構造体が自己参照型ポインタによってリング状に結合されているんですね。でもって各星たちは，自分の前に位置している星のあとを追いかけているわけです。そうすると，最初は画面上にランダムに配置されていた星が，だんだんと「わっか」の形に収束（？）していく，というわけです。原理は簡単でしょ？”

なんだそうです。……ごめんなさい，わかんないです（ああ，なさけないっ）。

ついでにだな，このプログラム，たとえばマウスカーソルを使って“肥料”をまいてさ，この宇宙を大きくしたり，いろんな形にできると育てがいがあっておもしろいかもね……って本当にシム宇宙になっちまうな。

ま，次回はぜひとも育てるのに人間が手を加えて楽しめるやつがいいな，と要望を出したところで今月は終わりにしましょう。それではまた来月っ！

```
101:         cmp.w   -4(a4),d5                    *左を調べる
102:         bne     pass
103:
104:         cmp.w   -V_1DOT*2(a4),d5             *上を調べる
105:         bne     pass
106:
107:         moveq.l #0,d0
108: pass:
109:         rts
110:
111:
112: rand:                                        *ここは細かい技を使っているので解説
113:                                              *はなし（ごめんなさい）
114:         dbra    d7,rn
115:         moveq.l #5,d7
116:         dc.w    __RAND                       *ここがボトルネックだ（ファンクショ
117:         rol.w   #4,d0                        *ンコールは全般に遅い）
118:         move.w  d0,d6                        *d6:乱数の保存（高速化）
119:         andi.w  #%00000000_00000110,d0
120:         lea     jmptbl,a0
121:         jmp     0(a0,d0.w)
122: rn:
123:         rol.w   #2,d6
124:         move.w  d6,d0
125:         andi.w  #%00000000_00000110,d0
126:         lea     jmptbl,a0
127:         jmp     0(a0,d0.w)
128:
129: jmptbl:
130:         bra.s   right                        *この行は消しても動くと思う
131:         bra.s   down
132:         bra.s   left
133:         bra.s   up
134:
135: right:
136:         cmp.w   4(a4),d5                     *右を調べて
137:         beq     rand                         *既に通路ならやり直し
138:
139:         addq.l  #2,a4
140:         move.w  d5,(a4)+                     *X68000のVRAMはこんな時に便利
141:         move.w  d5,(a4)
142:         rts
143:
144: down:
145:         cmp.w   V_1DOT*2(a4),d5              *下を調べて
146:         beq     rand                         *通路ならやり直し
147:
148:         adda.l  d3,a4                        *縦の時にはポストインクリメントは
149:         move.w  d5,(a4)                      *使えない
150:         adda.l  d3,a4
151:         move.w  d5,(a4)
152:         rts
153:
154: left:
155:         cmp.w   -4(a4),d5                    *上と同様
156:         beq     rand
157:
158:         move.w  d5,-(a4)
159:         move.w  d5,-(a4)
160:         rts
161:
162: up:
163:         cmp.w   -V_1DOT*2(a4),d5             *上と同様
164:         beq     rand
165:
166:         suba.l  d3,a4
167:         move.w  d5,(a4)
168:         suba.l  d3,a4
169:         move.w  d5,(a4)
170:         rts
171:
172:
173:
174:         .data
175:
176:         .even
177:
178: userstack:
179:         ds.l    1
180:
181: f_param:
182:         dc.w    0
183:         dc.w    0
184:         dc.w    767
185:         dc.w    511
186:         dc.w    pen
187:
188: f_param2:
189:         dc.w    1
190:         dc.w    1
191:         dc.w    765
192:         dc.w    509
193:         dc.w    paper
194:
195:
196:         .bss
197:         .even
198:         ds.w    30                           *これはスタックの予備
199: flowchk:
200:         ds.w    204800                       *スタックは大分節約したので
201: bigstack:                                    *もう少し削れるかも知れない
```

リスト2 wak.c

```
  1: /*-----------------------------------------------------------
  2:        宇宙の真理：超弦理論リング(おおウソ)Toy Program
  3:   [wak.c] 1990/12 T.Fukuhara/Yui.Kawahara/w.Marimo's Factory
  4:  -----------------------------------------------------------*/
  5: #include <stdio.h>
  6: #include <stdlib.h>
  7: #include <doslib.h>
  8: #include <iocslib.h>
  9: #include <graph.h>
 10: #include <basic.h>
 11: #define QA 100              /***星の数***/
 12: #define RATE 13.0           /***一回に動く量***/
 13: #define CO 300              /***移動の繰り返し回数***/
 14:
 15: typedef struct suna{
 16:         float x;
 17:         float y;
 18:         int c;
 19:         struct suna *nextp;
 20: }SAND;
 21:
 22: static SAND *sand,*s;
 23:
 24:
 25: void makesand();
 26: SAND *allocate();
 27: /*---------------------*/
 28: void g_init(void)
 29: {
 30:         int i,kara;
 31:         SAND *temp;
 32:
 33:         temp  = s;
 34:         while(temp != s->nextp){/***メモリー解放部***/
 35:                 sand = s->nextp;
 36:                 free(s);
 37:                 s = sand;
 38:         }
 39:         makesand();
 40:         for(i=15;i>=4;i--){
 41:                 for(kara = 0;kara<15000;kara++)
 42:                         contrast(i);
 43:         }
 44:         WIPE();
 45:         contrast(15);
 46: }
 47: void movesand(void)
 48: {
 49:         s->x += ((s->nextp->x) - (s->x))/RATE;
 50:         s->y += ((s->nextp->y) - (s->y))/RATE;
 51:         pset((int)s->x,(int)s->y,s->c);
 52: /*      printf("%d , %d, %d ¥n",(int)s->x,(int)s->y,s->c);     */
 53:         s=s->nextp;
 54: }
 55: void makesand(void)
 56: {
 57:         int i=QA;
 58:
 59:         s = allocate(i);
 60:         sand = s;
 61:         while(--i){
 62:                 s->nextp = allocate(i);
 63:                 s = s->nextp;
 64:         }
 65:         s->nextp = sand; /*ﾘﾝｸﾞ構造ﾅﾝﾀﾞﾅ!*/
 66: }
 67: /*---メモリーアロケート&初期値代入部---*/
 68: SAND *allocate(int i)
 69: {
 70:         SAND *temp;
 71:
 72:         temp = (SAND *)calloc(1,sizeof(SAND));
 73:         temp->x = (float)(rand() / 64);
 74:         temp->y = (float)(rand() / 64);
 75:         temp->c = hsv( 191*i/QA , (0x01 & i)?13:18, 31);
 76:         return(temp);
 77: }
 78: /*---------------------*/
 79: void main()
 80: {
 81:         int i=0;
 82:
 83:         screen(1,3,1,1);
 84:         console(0,31,0);
 85:         vpage(1);
 86:         apage(0);
 87:         B_CUROFF();
 88:         makesand();
 89:
 90:         while(!(INPOUT(0xFF) == 0x1b)){ /*ESC key で終了*/
 91:                 if(i++ < QA*CO)
 92:                         movesand();
 93:                 else{
 94:                         g_init();
 95:                         i=0;
 96:                 }
 97:         }
 98:         screen(2,0,1,1);
 99:         B_CURON();
100:         exit(0);
101: }
```

(で)のぱーてぃハンズ第2部――(その4)

前回（2カ月前のことですが）までで迷路というかの表示ルーチンができたわけですね。で，今月はこの迷路が入っている配列を作ってその中をさまよって，めでたく完成というわけでございます。

とはいっても，ダンジョンの画面パターンの表示ルーチンを作った時点でこのプログラムはほぼ完成したようなものだったりするんですけど。あとはほとんど力技でやってしまいます。

さあて，マッピング

それでっと。ふらふらとさまよいたいのですが，そのためにはまずダンジョンの地形というかマップのデータが必要になりますね。ということで1階分のマップデータを作ります。

ダンジョンなんですけど，今回はサンプルということもあるんでちょっと狭いけど20×20の広さにします（狭くていやだという人は自分で改良してね）。

で，仕様をちょこっと決めてしまいます。マップの情報なんですが，今回は壁，通路のほかに通り抜けられる壁なんていうのも作ろうと思います。赤い枠の壁がそれです。それからここで各種のイベント（たとえば“上に登る階段があります。登りますか？(Y/N)”とかね）を発生させるとなんとなくゲームっぽくていいなあ，なんて思ってしまう私だったりするわけです（実際には狭いマップをさまようだけであまり遊べるものじゃないんだけど）。

というわけで，ダンジョン1階分のマッピングをします。まずマップデータを作るのには何を使うんでしたっけ？　そう，配列ですよね。前回まで視界の範囲ということで配列を使いましたけど，ここでも同じように配列を使うんです。20×20の広さなので400個の変数を用意してやればいいわけですね。80～270行の，

int map(400)＝｛ 1,1,1,1,1,1,1,1,1……
　　　　　　……,1,1,1,1,1,1,1,1｝

がマップになります。ここの部分は自分でマッピングしてやってください。お好みのダンジョンが出来上がります（当たり前）。具体的には配列に入っているのは1が普通の壁，0が通路，2が通り抜けられる壁，5が階段イベントになっています。

さまよってしまうのだ

で，通路ができたと。あとはキー入力によって自分の位置を変えて，目に見えるものが変わったときに画面を描きかえるようにすれば完成なわけです。制作もこれで最後になるのでこれがメインルーチンということにします（おお，トップダウン）。

メインルーチンなんだけど，まずキー入力で2,4,6,8のキーが押されるわけです。で，たとえば8のキーを押された場合，前進するということですよね。ということは，まず目の前が壁でないことを確認して（壁だったら進めません，当然）から，いま向いている方向に座標を増減します。これで，自分は動いたと。

次は画面のほうですね。前回までに視界の範囲ということで作った3×4個の配列がありますね。この内容はどうなるでしょう？　そうですよね。自分が1歩前に進むんだから1コマ前にずれた3×4の範囲の内容になるんですよね。なんだ簡単じゃん，ということで，このプログラムでは素直にひとつずれた先の内容を12個全部，視界の範囲の配列に1回1回転送してるんです。遅くなりそうだけど，組んでみたら思ったほど遅くならなかったんでそうしてしまいました。考え方としては簡単ですからこれでいいですよね（気にくわない人は改良してね，と何度も繰り返しいってしまう私なのだった）。

で，キー入力が4,6,2であった場合には自分の場所は変えずに方向を90度，あるいは180度回転することにしています。

ちなみに8で移動したときにevent( )ルーチンが自分のいるマップから各種のイベントを判定して，“階段があります”とか（というよりこれしか作ってない）教えてくれます。しかし，

switch〜caseでがんがん場合分けしたからリストが長い長い。ちょっと反省。

……というわけで，3Dダンジョンプログラムが完成しました！　ぱーてぃハンズ第2部「ダンジョンなのである」はついにここで幕を閉じてしまうのであります。いやあ，めでたし，めでたしと。

ううむ。本当に3Dダンジョンの表示の基礎だけで終わってしまったなあ。ま，表示ルーチンもあることだし，皆さんのほうで10階分くらいマップを用意して(このあと階段イベントで配列を入れ替えるのね)，ついでに戦闘シーンを付け足すとか，迷路を自動生成させるプログラムを作って自分でさまよってみる（ちょうど今月似たようなプログラムがあることだし）とか，ぜひいろいろと活用してみていただきたいのであります。

で，このコーナーなんですけど，結構好評みたいで（……うるうる。ハガキありがとうございます。読者様は神様です）このまま第3部へとなだれこんでしまいます。

とはいっても全然ネタなんか考えてなかったんだよな（このへんはスピリッツの猿まんをほうふつとさせる)。ええと，1本目がアクションゲーム，2本目がダンジョンの表示ルーチンだから……。3本目は思い切って思考ゲームなんかも面白いかもしれません。2人対戦型の思考ゲームを作って，ついでに思考ルーチンを作ってしまおうかなあという大風呂敷を広げてみたりして。どうだ，思考ルーチンだぞ。ううん，できるといいなぁ。

おっと，一応ここで1本釘をさしておこう。あくまでも，このコーナーはショートプロのページですんでね。思考ゲームということになってもあんまりたいしたものはできないと思いますよ。というわけで，あんまり過大な期待は抱かないようにね。

では，またこのOh!Xで。第2部完！(と字幕を出して去る)

リスト

```
  80 int map(400)=( 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
  90               ,1,0,0,0,0,0,0,5,2,0,0,0,0,0,0,0,0,0,1
 100               ,1,1,5,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1
 110               ,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1
 120               ,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1
 130               ,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1
 140               ,1,1,5,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1
 150               ,1,1,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
 160               ,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
 170               ,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
 180               ,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
 190               ,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
 200               ,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
 210               ,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
 220               ,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
 230               ,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
 240               ,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
 250               ,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
 260               ,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
 270               ,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)
 300 setary(myxy,dir)
 310 /*メインルーチン*/
 320 while(1)
 330 mflg=moveme()
 340 /*print mflg*/
 350 if mflg<>0 then setary(myxy,dir):drawmaze()
 360 event()
 370 endwhile
 390 func event()
 400 switch map(myxy)
 410         case 1:isinonaka():break
 420         case 5:kaidan()    :break
 430 endswitch
 440 endfunc
 450 func isinonaka()
 460 cls:locate 0,24
 470 print"石の中にいる"
 480 while(1):endwhile
 490 endfunc
 500 /*階段がある*/
 510 func kaidan()
 520 str a
 530 int i=0
 540 while(i=0)
 550 cls:locate 0,24:print"階段があります上りますか(Y/N)"
 560 a=inkey$(0)
 570 if a="Y" or a="y" then stop
 580 if a="N" or a="n" then i=1
 590 print a
 600 endwhile
 610 endfunc
 620 func moveme()
 630 str  astr
 640 astr=inkey$(0)
 650 switch asc(astr)
 660         case '8':return(goahead()):break
 670         case '2':turnback():return(1):break
 680         case '4':turnleft():return(1):break
 690         case '6':turnright():return(1):break
 700         default :return(0):break
 710 endswitch
 720 endfunc
 730 func goahead()
 740 int  dxy
 750 switch dir
 760         case 0:dxy=+1:break
 770         case 1:dxy=-1:break
 780         case 2:dxy=20:break
 790         case 3:dxy=-20:break
 800 endswitch
 810 if map(myxy+dxy)<>1 then myxy=myxy+dxy:return(1) else beep
:return(0)
 820 endfunc
 830 func turnback()
 840 switch dir
 850         case 0:dir=1:break
 860         case 1:dir=0:break
 870         case 2:dir=3:break
 880         case 3:dir=2:break
 890 endswitch
 900 endfunc
 910 func turnleft()
 920 switch dir
 930         case 0:dir=3:break
 940         case 1:dir=2:break
 950         case 2:dir=0:break
 960         case 3:dir=1:break
 970 endswitch
 980 endfunc
 990 func turnright()
1000 switch dir
1010         case 0:dir=2:break
1020         case 1:dir=3:break
1030         case 2:dir=1:break
1040         case 3:dir=0:break
1050 endswitch
1060 endfunc
1140 func setary(xy,d)
1150 /*バッファにメイズデータをセット*/
1160 switch d
1170     case 0:setar0(xy):break
1180     case 1:setar1(xy):break
1190     case 2:setar2(xy):break
1200     case 3:setar3(xy):break
1210 endswitch
1220 endfunc
1230 func setar0(xy)
1240 /*d   =0    X  +方向*/
1250 int w1
1260 for w1=0 to 3
1270 if xy-20+w1<401 then ary(w1)=map(xy-20+w1)
1280 next
1290 for w1=0 to 3
1300 if xy+w1<401 then ary(w1+4)=map(xy+w1)
1310 next
1320 for w1=0 to 3
1330 if xy+20+w1<401 then ary(w1+8)=map(xy+20+w1)
1340 next
1350 setex()
1360 endfunc
1370 func setar1(xy)
1380 /*    =2    X  -方向*/
1390 int w1
1400 for w1=0 to 3
1410 if xy+20-w1>-1 then ary(w1)=map(xy+20-w1)
1420 next
1430 for w1=0 to 3
1440 if xy-w1>-1  then ary(w1+4)=map(xy-w1)
1450 next
1460 for w1=0 to 3
1470 if xy-20-w1>-1  then ary(w1+8)=map(xy-20-w1)
1480 next
1490 setex()
1500 endfunc
1510 func setar2(xy)
1520 /*    =2    Y  +方向*/
1530 int w1
1540 for w1=0 to 3
1550 if xy+1+w1*20<401 then ary(w1)=map(xy+1+w1*20)
1560 next
1570 for w1=0 to 3
1580 if xy+w1*20<401 then ary(w1+4)=map(xy+w1*20)
1590 next
1600 for w1=0 to 3
1610 if xy-1+w1*20<401 then ary(w1+8)=map(xy-1+w1*20)
1620 next
1630 setex()
1640 endfunc
1650 func setar3(xy)
1660 /*    =3    Y  -方向*/
1670 int w1
1680 for w1=0 to 3
1690 if xy-1-w1*20>-1  then ary(w1)=map(xy-1-w1*20)
1700 next
1710 for w1=0 to 3
1720 if xy-w1*20>-1  then ary(w1+4)=map(xy-w1*20)
1730 next
1740 for w1=0 to 3
1750 if xy+1-w1*20>-1  then ary(w1+8)=map(xy+1-w1*20)
1760 next
1770 setex()
1780 endfunc
1790 func setex() /*4以上はイベントなので0扱いにする*/
1800 int i
1810 for i=0 to 11
1820 if ary(i)>4 then ary(i)=0
1830 next
1840 for i=0 to 2
1850 /*print ary(i*4);ary(i*4+1);ary(i*4+2);ary(i*4+3) */
1860 next
1870 endfunc
```

マシン語カクテル in Z80's Bar

第20回——事故の前にブレーキング——

シナリオ：金子俊一
特別監修：浦川博之
イラスト：山田純二

光君がやっとプログラムを作ってツケを払ってくれる今月のマシン語カクテル。ものはデバッガということですが，主にブレイクポイントのお話になっています。暴走という恐怖がつきまとうマシン語のデバッグには絶対必要ですよね。

♪カラン，コロ～ン
源光（以下光）：こんにちは。
マスター（以下M）：おや，その顔色では危ぶまれていた進級はセーフだったようですね。
光：なにいってんですか。私は学業優秀なんですよ。
M：本当かねえ。あれだけ学校サボっといて。
光：まかせてくださいよ。ひとのノートで評価Aを取る男って有名なんですよ。
M：それって，いちばん嫌われるタイプじゃないですか。
光：その分，プログラムの宿題は全部回ってきますから。
ようこ（以下Yo）：私も手伝ってもらったことがあるような，ないような。
長老（以下老）：わしゃ，ようこちゃんがプログラム組むところを見たことないんじゃが。
M：そういえば，いつもプログラムの宿題は光君がやってますねぇ。
Yo：失礼ね，これでもときどきはプログラミングしてるのよ。ねえ，光君。
光：さあ……？
Yo：ひっどーい。それじゃあ，私の作ったプログラムでぎゃふんといわせてあげるわよ。
老：どれどれ，お手なみ拝見といきますかの。
Yo：えっと，プログラムをロードしてと……。えい，実行！

小さな防衛線

Z80（特別出演）：ぎゃふん。
光：あ～あ，暴走しちゃった。
老：なんだかコンピュータが悲鳴をあげておったのう。
Yo：どうせ初心者ですよーだ。だいたい，一度実行したら止まらないマシン語がいけないのよ。
光：止まらないワケでもないよ。
Yo：そんなのウソよ。
老：うむ，世間では暴走しても1/2の確率で「アドレスエラーが発生しました」って騒ぐやつもおるしのう（笑）。
M：ひょっとして，よく知っているやつですね。
光：あれはシステムが強力だからできるワザですけどね。一般的にZ80では小細工を使わないと無理でしょうね。
Yo：どうするの？
光：デバッガを使うのがお手頃だろうね。
Yo：でばっがあ？
光：うん，Oh!Xだったら，1988年の4月号に載ってるよ。ちょうど3年前のものになるんだけど。
Yo：そんな古いの持ってないわ。だいたい私がマシン語始めたのはおととしの夏よ。
老：ふぉっふぉっふぉ，青いのう。
Yo：わかったから教えて。
光：デバッガのブレイクポイントっていう機能を使うんだ。
Yo：ぶれいくぽいんと？
光：そう，マシン語を止める魔法のおマジナイさ。
M：タネはあるんでしょ。
老：マスターもするどいのう。
光：普段はあんまりお世話にならないRST命令を使うんですよ。
M：ほう。
老：RST命令は特定の番地をコールする1バイト命令じゃな。
光：はい，1バイトっていうのがミソなんですよ。特にRST 38なんかはマシンコードが$FFだからメモリダンプしててもわかりやすいし，手頃なんですよね。
老：小さな防衛線じゃのう。
Yo：どうやって使うの？
光：だから，止めたいアドレスに$FFを書き込む。もちろん，元のデータはちゃんととっておく。
老：あとで元に戻すためじゃな。
光：そういうこと。プログラムを実行している最中にRST命令があると，CPUは$0038番地をコールするんだ。ただし，MZ80K/80C/1200/700/1500では，$1038番地をコールするようだね。そこにレジスタを表示するプログラムを入れておけばいいわけ。
老：実際には$00XX番地付近はシステム関係のワークがあったりするので，$0038番地にはJP命令を入れておくのじゃ。
光：まあ，わかってしまえば仕掛けは簡単なんですよ。
Yo：わかんない。
光：それではプログラムを見ながら話を進めましょう。
M：そっそれじゃ，たまりにたまっていたツケをとうとう払ってくれるんですね？
光：やだなマスター，涙目になっちゃって。いままでは家のX1turboが壊れていて，プログラムが作れなかっただけですよ。
M：それじゃ今回はどうして作れるんですか？
光：いや，友人のH.K.君からX1turboをむりやり借りたんですよ。はっはっは。
老：持つべきものは友達じゃのう。
光：それでは，カチャカチャ……。
老：これは結構長くかかりそうだのう。

時間よ，止まれ

2日後……。
光：終わった終わった。
M：ごくろうさま。はい，つぶつぶ入り赤

マムシドリンク。

Yo：ちょっとマスター，とびきりヘンなものを渡さないでよ。

光：ゴクゴク。うっ鼻血出そう。さて，なんでも説明しますよ。

老：どんなものを作ったのかのう？

光：まあ，デバッガと呼ぶにはちょっとおおげさだけど，ブレイクポイントの機能，メモリエディット，ダンプ，ジャンプ，ブロック転送ができるものかな。

老：おお，あとはディスアセンブルとメモリサーチがつけばデバッガとして成り立つわけじゃな。

光：トレースやファイル操作までつけば完璧なんでしょうけどね。サイズが小さいほうが使い勝手はいいと思いますから。

Yo：ブレイクポイントのあたりを詳しく説明して。

光：はいな。リストの後ろのほうにあるDISPLAYっていうサブルーチンがあるでしょ。これがミソ。AF,BC,DE,HL,AF',BC',DE',HL',IX,IY,SP,PCの各レジスタを表示するルーチンなんだ。

Yo：ブレイクポイントが働いたときにここにくるわけね。

光：そう。スタックポインタをうまく利用しているんだ。S-OSではHLレジスタを表示するサブルーチンが用意されているから，レジスタを一度スタックに入れてからHLでPOPして，表示している。あと，フラグレジスタ表示に使っているBIN$というルーチンは面白いと思うよ。

Yo：なにかポイントでもあるの？

光：Lレジスタの各ビットを見て，0だったら0，1だったら1と表示するだけなんだけど，条件分岐を使っていないんだ。

Yo：リストを見ればだいたいわかるわね。

M：ADC命令の正統的な(?)使い方ですね。

光：ブレイクポイントの管理はBP～というルーチンでやってる。セットするときとリセットするときで共通に使えそうなルーチンを先に作っておいたんだ。

Yo：BPのステッカーと関係あるの？

光：全然なーい。

老：リストではBPSEARCHとBPLOAD，BPLISTといったところじゃな。

光：そうですね。BPSEARCHではCレジスタに0を入れてコールすると空きワークエリアを，Cレジスタに1を入れてコールすると使っているワークエリアを探すんだ。BPLOADではブレイクポイントのアドレスをBCレジスタに入れてくれる。BPLISTは設定されているブレイクポイントの一覧を表示する。

老：そこまで説明すればわかるじゃろう。注意する点はあるのかのう。

光：同じアドレスを設定できないようにチェックする必要がありますね。それから$FFを入れる前にそこの値を待避させておくこと。あとはワークエリアの大きさを超えなければ大丈夫。これで安心してマシン語も止めることができますよ。

M：よかった，よかった。

老：このソースリストではオブジェクトが8000$_H$番地に生成されるようじゃのう。

光：ソースから打ち込んだときにはアセンブル後に5000$_H$番地に転送する必要がありますね。

老：アセンブルしたあとに，OFFSETの行を外して，「ORG $5000」を「ORG $9000」に変えて，もう一度アセンブル。そして，S-OSに戻って，

#J9000

コマンドの説明

○!

S-OSのモニタにジャンプします

ex)　!

○EXXXX

XXXX番地のメモリをエディットします。数値を続けて書くこともできます。"；"のあとに書いた文字はASCII文字としてそのままメモリに格納されます。

ex)　EA000

○DXXXX

XXXX番地を先頭とするメモリを$80バイトだけダンプします。アドレスを省略すると前回の続きをダンプします。ダンプして出力したデータはそのままエディットできます。

ex)　D8000

○JXXXX

XXXX番地をコールします。プログラム中のRET命令で帰ってきます。

ex)　J3000

○TXXXX YYYY ZZZZ

XXXX番地からYYYY番地までのデータをZZZZ番地に転送します。指定方法を間違えるか，番地を設定しなかった場合には簡単なヘルプを出力します。

ex)　T8000 80FF 9000

○BXXXX

ブレイクポイントをXXXX番地に設定します。プログラムを実行中にXXXX番地の命令を実行しようとすると，AF,BC,DE,HL,AF',BC',DE',HL',IX,IY,SP,PCを表示して，このプログラムのモニタが起動します。ブレイクポイントは最大8個まで登録できます。ただし，同じアドレスを登録することはできません。さらに，コンティニューはできませんので注意してください。アドレスを省略すると，現在設定しているブレイクポイントのアドレスを表示します。なお，本機能はZ80の命令のRST 38($FF)を利用しています。よって，ブレイクポイントはオペコードのアドレスを指定してください。たとえば，

```
8000  06 01   : LD B,1
```

のようなときにブレイクポイントを$8001に設定すると，

```
8000  06 FF   : LD B,FF
```

となり，ブレイクポイントは設定されません。この場合にはブレイクポイントを$8000に設定してください。

```
8000  FF 01   : RST 38  LD BC,
```

これならブレイクポイントが働きます。

ex)　B5613

○&XXXX

設定したブレイクポイントを解除します。アドレスを省略すると，現在設定しているブレイクポイントのアドレスを表示します。なお，ブレイクポイントを設定したまま"!"コマンドで終了すると，指摘されます。ブレイクポイントを解除したいときは，そのままプログラムを立ち上げてください。初期化をしませんのでそこで&コマンドを使えば大丈夫です。

ex)　&5613

○M

各機種のマシン語モニタを起動します。モニタを抜けるとS-OS上に戻ります。

ex)　M

○#

プリンタのON/OFFのスイッチです。トグルになっています。ダンプをするときにチェックされます。

ex)　#

とすれば，このプログラムが立ち上がるじゃろう。あとはこのモニタ上のTコマンドで，

　A＞T8000 83FF 5000

とすれば，5000H番地に転送できるのう。

光：入力ミスがないのが前提ですね。けっこう危ないと思うけど。ちなみにプログラムはちょうど１Kバイトになっていますから，参考にするといいでしょうね。

老：それじゃ試しに動かしてみようかの。

Yo：どれどれ，あっ本当に止まるのね。さすが光君ね。

光：ちゃんと止まりますよ。

M：一時停止無視は２点取られるそうですからね。

老：罰金は7,000円だそうじゃのう。

光：なにをボケまくって１年以上も前の

をしてるんですか。

M：なあに，場所がお台場だったなんて

にもいってませんよ。

老：横には誰が乗っていたのかのう。ふ

っふぉっふぉ。

Yo：みんなのイジワル！

―つづく

リスト１

```
0000                 1 ;      Break Pointer
0000                 2 ;
0000                 3 ;          by Hikaru Minamoto
0000                 4
0000                 5          OFFSET  $8000-$5000
5000                 6          ORG     $5000
5000                 7
5000                 8 ;Label           Address   Break
5000                 9
5000                10 #DSK     EQU     $1F5D   ; system work
5000                11 #STKAD   EQU     $1F6C   ;         :
5000                12 #KBFAD   EQU     $1F76   ;         :
5000                13 #PRCNT   EQU     $1F7A   ;         :
5000                14 #LPSW    EQU     $1F7C   ;         :
5000                15 #MON     EQU     $1F8E   ; nothing
5000                16 #HLHEX   EQU     $1FB2   ; AF,DE+4,HL
5000                17 #2HEX    EQU     $1FB5   ; AF,DE+2
5000                18 #HEX     EQU     $1FB8   ; AF
5000                19 #PRTHL   EQU     $1FBE   ; AF
5000                20 #PRTHX   EQU     $1FC1   ; AF
5000                21 #BELL    EQU     $1FC4   ; AF
5000                22 #GETL    EQU     $1FD3   ; AF
5000                23 #LPTOF   EQU     $1FD6   ; nothing
5000                24 #LPTON   EQU     $1FD9   ; nothing
5000                25 #MPRINT  EQU     $1FE2   ; AF,DE
5000                26 #LTNL    EQU     $1FEE   ; nothing
5000                27 #PRINTS  EQU     $1FF1   ; F
5000                28 #PRINT   EQU     $1FF4   ; F
5000                29 #VER     EQU     $1FF7   ; HL
5000                30 #HOT     EQU     $1FFA   ; nothing
5000                31 #RDVSW   EQU     $2024   ; A
5000                32 #SDVSW   EQU     $2027   ; AF
5000                33
5000                34
5000                35 COLD
5000 CD F7 1F       36          CALL    #VER
5003 7C             37          LD      A,H
5004 21 38 00       38          LD      HL,$0038
5007 FE 03          39          CP      3
5009 30 02          40          JR      NC,COLD2
500B CB E4          41          SET     4,H     ; MZ-80K,ETC =$1038
500D                42 COLD2
500D 3E C3          43          LD      A,$C3
500F 77             44          LD      (HL),A
5010 11 83 52       45          LD      DE,DISPLAY
5013 23             46          INC     HL
5014 73             47          LD      (HL),E
5015 23             48          INC     HL
5016 72             49          LD      (HL),D
5017                50          ;
5017 CD E2 1F       51          CALL    #MPRINT
501A 0C             52          DB      $0C
501B 3C 3C 3C       53          DM      "<<<<< Break Pointer"
501E 3C 3C 20
5021 42 72 65
5024 61 6B 20
5027 50 6F 69
502A 6E 74 65
502D 72
502E 20 76 65       54          DM      " version 0.68 >>>>>"
5031 72 73 69
5034 6F 6E 20
5037 30 2E 36
503A 38 20 3E
503D 3E 3E 3E
5040 3E
5041 0D 00          55          DB      $0D,$00
5043                56          ;
5043                57 HOT
5043 CD D6 1F       58          CALL    #LPTOF
5046 ED 7B 6C       59          LD      SP,(#STKAD)
5049 1F
504A CD 24 20       60          CALL    #RDVSW  ; A,(#DSK)
504D 32 6B 50       61          LD      (HOT2),A
5050 CD F4 1F       62          CALL    #PRINT
5053 3E 3E          63          LD      A,">"
5055 CD F4 1F       64          CALL    #PRINT
5058 ED 5B 76       65          LD      DE,(#KBFAD)
505B 1F
505C CD D3 1F       66          CALL    #GETL
505F 2A 7A 1F       67          LD      HL,(#PRCNT)
5062 36 00          68          LD      (HL),0
5064 21 43 50       69          LD      HL,HOT
5067 E5             70          PUSH    HL
5068 CD 73 50       71          CALL    SPSEA
506B                72 HOT2
506B 41 3E          73          DM      "A>"
506D 00             74          DB      0
506E 30 25          75          JR      NC,COMMAND
5070 C3 21 51       76          JP      MEMSET
5073                77 SPSEA
5073 E3             78          EX      (SP),HL
5074 C5             79          PUSH    BC
5075 06 00          80          LD      B,0
5077                81 SPSE1
5077 7E             82          LD      A,(HL)
5078 B7             83          OR      A
5079 28 0F          84          JR      Z,SPSE4
507B 1A             85          LD      A,(DE)
507C BE             86          CP      (HL)
507D 20 0F          87          JR      NZ,SPSE5
507F 23             88          INC     HL
5080 13             89          INC     DE
5081 04             90          INC     B
5082 18 F3          91          JR      SPSE1
5084                92 SPSE2
5084 04             93          INC     B
5085 13             94          INC     DE
5086                95 SPSE3
5086 1B             96          DEC     DE
5087 10 FD          97          DJNZ    SPSE3
5089 37             98          SCF
508A                99 SPSE4
508A C1            100          POP     BC
508B 23            101          INC     HL
508C E3            102          EX      (SP),HL
508D C9            103          RET
508E               104 SPSE5
508E 7E            105          LD      A,(HL)
508F B7            106          OR      A
5090 28 F2         107          JR      Z,SPSE2
5092 23            108          INC     HL
5093 18 F9         109          JR      SPSE5
5095               110          ;
5095               111 COMMAND
5095 1A            112          LD      A,(DE)
5096 13            113          INC     DE
5097 B7 C8         114          IF A=0   RET
5099 FE 21 28      115          IF A="!" JR CLOSE
509C 40
509D FE 4A 28      116          IF A="J" JR JUMP
50A0 78
50A1 FE 4D 28      117          IF A="M" JR MON
50A4 79
50A5 FE 45 28      118          IF A="E"・JR MEMSET
50A8 78
50A9 FE 23 CA      119          IF A="#" JP PRNSET
50AC 75 51
50AE FE 54 CA      120          IF A="T" JP TRANS
50B1 A5 51
50B3 FE 42 CA      121          IF A="B" JP BPSET
50B6 EB 51
50B8 FE 26 CA      122          IF A="&" JP BPCLR
50BB 22 52
50BD FE 44 CA      123          IF A="D" JP DUMP
50C0 B0 53
50C2 FE 46 CA      124          IF A="F" JP FIND        ;
50C5 FF 53
50C7 FE 4C CA      125.         IF A="L" JP DISASM      ;
50CA FF 53
50CC FE 56 CA      126          IF A="V" JP DVSET       ;
50CF FF 53
50D1 FE 52 CA      127          IF A="R" JP READ        ;
50D4 FF 53
50D6 FE 57 CA      128          IF A="W" JP WRITE       ;
50D9 FF 53
50DB 18 B8         129          JR      COMMAND
50DD               130 CLOSE ;*
50DD CD D6 1F      131          CALL    #LPTOF
50E0 0E 01         132          LD      C,1
50E2 CD 72 52      133          CALL    BPSEARCH
50E5 DA FA 1F      134          JP      C,#HOT
50E8 CD E2 1F      135          CALL    #MPRINT
50EB 43 61 75      136          DM      "Caution ! "
50EE 74 69 6F
50F1 6E 20 21
50F4 20
50F5 42 72 65      137          DM      "Break pointer"
50F8 61 6B 20
50FB 70 6F 69
50FE 6E 74 65
5101 72
5102 20 69 73      138          DM      " is still setting."
5105 20 73 74
5108 69 6C 6C
510B 20 73 65
510E 74 74 69
5111 6E 67 2E
5114 0D 00         139          DB      $0D,$00
5116 C3 FA 1F      140          JP      #HOT
5119               141
5119               142 JUMP ;*
5119 CD B2 1F      143          CALL    #HLHEX
511C D8            144          RET     C
511D E9            145          JP      (HL)
511E               146
511E               147 MON ;*
511E C3 8E 1F      148          JP      #MON
5121               149
5121               150 MEMSET ;*
5121 CD B2 1F      151          CALL    #HLHEX
5124 D8            152          RET     C
```

```
5125                153 EDIT
5125 CD 47 51       154          CALL     EDIT2
5128 CD BE 1F       155          CALL     #PRTHL
512B CD F1 1F       156          CALL     #PRINTS
512E                157          ;
512E 7E             158          LD       A,(HL)  ;*
512F CD C1 1F       159          CALL     #PRTHX
5132 3E 1D          160          LD       A,$1D   ; CUR_LEFT
5134 CD F4 1F       161          CALL     #PRINT
5137 CD F4 1F       162          CALL     #PRINT  ;*
513A                163          ;
513A ED 5B 76       164          LD       DE,(#KBFAD)
513D 1F
513E CD D3 1F       165          CALL     #GETL
5141 CD B2 1F       166          CALL     #HLHEX
5144 30 DF          167          JR       NC,EDIT
5146 C9             168          RET
5147                169 EDIT2
5147 1A             170          LD       A,(DE)
5148 B7             171          OR       A
5149 C8             172          RET      Z
514A FE 3B          173          CP       ";"
514C 28 18          174          JR       Z,EDIT6
514E FE 20          175          CP       " "
5150 28 06          176          JR       Z,EDIT4
5152                177 EDIT3
5152 CD B8 1F       178          CALL     #HEX
5155 D8             179          RET      C
5156 18 01          180          JR       EDIT4+1
5158                181 EDIT4
5158 13             182          INC      DE
5159 1A             183          LD       A,(DE)
515A FE 3B          184          CP       ";"
515C 28 08          185          JR       Z,EDIT6
515E CD B5 1F       186          CALL     #2HEX
5161 D8             187          RET      C
5162                188 EDIT5
5162 77             189          LD       (HL),A
5163 23             190          INC      HL
5164 18 E1          191          JR       EDIT2
5166                192 EDIT6
5166 13             193          INC      DE
5167 1A             194          LD       A,(DE)
5168 13             195          INC      DE
5169 18 F7          196          JR       EDIT5
516B                197
516B                198 PRN
516B 3A A4 51       199          LD       A,(@PRFLG)
516E B7             200          OR       A
516F C2 D9 1F       201          JP       NZ,#LPTON
5172 C3 D6 1F       202          JP       #LPTOF
5175                203
5175                204 PRNSET ;*
5175 3A A4 51       205          LD       A,(@PRFLG)
5178 B7             206          OR       A
5179 28 15          207          JR       Z,PRNSET2
517B CD E2 1F       208          CALL     #MPRINT
517E 50 72 69       209          DM       "Printer OFF"
5181 6E 74 65
5184 72 20 4F
5187 46 46
5189 0D 00          210          DB       $0D,$00
518B AF             211          XOR      A
518C 32 A4 51       212          LD       (@PRFLG),A
518F C9             213          RET
5190                214 PRNSET2
5190 CD E2 1F       215          CALL     #MPRINT
5193 50 72 69       216          DM       "Printer ON"
5196 6E 74 65
5199 72 20 4F
519C 4E
519D 0D 00          217          DB       $0D,$00
519F 3C             218          INC      A
51A0 32 A4 51       219          LD       (@PRFLG),A
51A3 C9             220          RET
51A4                221 @PRFLG
51A4 00             222          DS       1
51A5                223
51A5                224 TRANS ;*
51A5 1A             225          LD       A,(DE)
51A6 FE 20          226          CP       " "
51A8 20 03          227          JR       NZ,TRANS2
51AA 13             228          INC      DE
51AB 18 F8          229          JR       TRANS
51AD                230 TRANS2
51AD CD B2 1F       231          CALL     #HLHEX
51B0 38 1F          232          JR       C,TRANS3
51B2 E5             233          PUSH     HL
51B3                234          ;
51B3 13             235          INC      DE
51B4 CD B2 1F       236          CALL     #HLHEX
51B7 C1             237          POP      BC
51B8 38 17          238          JR       C,TRANS3
51BA B7             239          OR       A
51BB ED 42          240          SBC      HL,BC
51BD 23             241          INC      HL
51BE C5             242          PUSH     BC       ; START
51BF E5             243          PUSH     HL       ; COUNT
51C0                244          ;
51C0 13             245          INC      DE
51C1 CD B2 1F       246          CALL     #HLHEX
51C4 D1             247          POP      DE       ; COUNT
51C5 C1             248          POP      BC       ; START
51C6 38 09          249          JR       C,TRANS3
51C8 C5             250          PUSH     BC       ; START
51C9 D5             251          PUSH     DE       ; COUNT
51CA E5             252          PUSH     HL       ; NEW
51CB                253          ;
51CB D1             254          POP      DE
51CC C1             255          POP      BC
51CD E1             256          POP      HL
51CE ED B0          257          LDIR
51D0 C9             258          RET
51D1                259 TRANS3
51D1 CD E2 1F       260          CALL     #MPRINT
51D4 41 3E 54       261          DM       "A>T $START $END $NEW"
51D7 20 24 53
```

```
51DA 54 41 52
51DD 54 20 24
51E0 45 4E 44
51E3 20 24 4E
51E6 45 57
51E8 0D 00          262          DB       $0D,$00
51EA C9             263          RET
51EB                264
51EB                265 BPSET ;*
51EB CD B2 1F       266          CALL     #HLHEX   ; HL=ADDRESS
51EE 38 5A          267          JR       C,BPLIST
51F0                268          ;
51F0 B7             269          OR       A
51F1 06 08          270          LD       B,8
51F3 0E 01          271          LD       C,1      ; FLAG=1
51F5 11 8F 53       272          LD       DE,@BP
51F8                273 BPSET2
51F8 CD 78 52       274          CALL     BPSER2
51FB 38 0F          275          JR       C,BPSET3
51FD C5             276          PUSH     BC
51FE CD 69 52       277          CALL     BPLOAD
5201 B7             278          OR       A
5202 E5             279          PUSH     HL
5203 ED 42          280          SBC      HL,BC
5205 E1             281          POP      HL
5206 C1             282          POP      BC
5207 CA C4 1F       283          JP       Z,#BELL  ; SAME ADDRESS
520A 18 EC          284          JR       BPSET2
520C                285 BPSET3
520C 0E 00          286          LD       C,0
520E CD 72 52       287          CALL     BPSEARCH
5211 DA C4 1F       288          JP       C,#BELL  ; BP NUMBER > 8
5214 7E             289          LD       A,(HL)
5215 EB             290          EX       DE,HL
5216 34             291          INC      (HL)     ; SET FLAG
5217 23             292          INC      HL
5218 73             293          LD       (HL),E
5219 23             294          INC      HL
521A 72             295          LD       (HL),D
521B 23             296          INC      HL
521C 77             297          LD       (HL),A
521D EB             298          EX       DE,HL
521E                299          ;
521E 36 FF          300          LD       (HL),$FF         ; SET BP
5220                301          ;
5220 18 28          302          JR       BPLIST
5222                303
5222                304 BPCLR ;*
5222 CD B2 1F       305          CALL     #HLHEX
5225 38 23          306          JR       C,BPLIST
5227                307          ;
5227 B7             308          OR       A
5228 06 08          309          LD       B,8
522A 0E 01          310          LD       C,1
522C 11 8F 53       311          LD       DE,@BP
522F                312 BPCLR2
522F CD 78 52       313          CALL     BPSER2
5232 38 16          314          JR       C,BPLIST
5234 C5             315          PUSH     BC
5235 CD 69 52       316          CALL     BPLOAD
5238 B7             317          OR       A
5239 E5             318          PUSH     HL
523A ED 42          319          SBC      HL,BC
523C E1             320          POP      HL
523D C1             321          POP      BC
523E 28 02          322          JR       Z,BPCLR3
5240 18 ED          323          JR       BPCLR2
5242                324 BPCLR3
5242 1B             325          DEC      DE
5243 1A             326          LD       A,(DE)
5244 77             327          LD       (HL),A
5245 1B             328          DEC      DE
5246 1B             329          DEC      DE
5247 1B             330          DEC      DE
5248 AF             331          XOR      A
5249 12             332          LD       (DE),A   ; FLAG=0
524A                333
524A                334 BPLIST
524A 06 08          335          LD       B,8
524C 0E 01          336          LD       C,1
524E 11 8F 53       337          LD       DE,@BP
5251                338 BPLIS2
5251 CD 78 52       339          CALL     BPSER2
5254 38 0F          340          JR       C,BPLIS3
5256 C5             341          PUSH     BC
5257 CD 69 52       342          CALL     BPLOAD
525A C5             343          PUSH     BC
525B E1             344          POP      HL
525C C1             345          POP      BC
525D CD BE 1F       346          CALL     #PRTHL
5260 CD F1 1F       347          CALL     #PRINTS
5263 18 EC          348          JR       BPLIS2
5265                349 BPLIS3
5265 CD EE 1F       350          CALL     #LTNL
5268 C9             351          RET
5269                352
5269                353 BPLOAD
5269 13             354          INC      DE
526A EB             355          EX       DE,HL
526B 4E             356          LD       C,(HL)
526C 23             357          INC      HL
526D 46             358          LD       B,(HL)
526E EB             359          EX       DE,HL
526F 13             360          INC      DE
5270 13             361          INC      DE       ; DE=NEXT FLAG
5271 C9             362          RET
5272                363 BPSEARCH
5272 B7             364          OR       A        ; RES CY
5273 06 08          365          LD       B,8
5275 11 8F 53       366          LD       DE,@BP
5278                367 BPSER2
5278 1A             368          LD       A,(DE)   ; A=FLAG
5279 B9             369          CP       C        ; C=0 or C=1
527A C8             370          RET      Z
527B 13             371          INC      DE
527C 13             372          INC      DE
527D 13             373          INC      DE
```

```
527E 13          374         INC     DE
527F 10 F7       375         DJNZ    BPSER2
5281 37          376         SCF
5282 C9          377         RET
5283             378
5283             379 DISPLAY
5283 FD E5       380         PUSH    IY
5285 DD E5       381         PUSH    IX
5287 D9          382         EXX
5288 08          383         EX      AF,AF'
5289 E5          384         PUSH    HL      ; ｳﾗ ﾚｼﾞｽﾀ
528A D5          385         PUSH    DE
528B C5          386         PUSH    BC
528C F5          387         PUSH    AF
528D F5          388         PUSH    AF      ; for ﾌﾗｸﾞ ﾚｼﾞｽﾀ
528E D9          389         EXX
528F 08          390         EX      AF,AF'
5290 E5          391         PUSH    HL      ; ｵﾓﾃ ﾚｼﾞｽﾀ
5291 D5          392         PUSH    DE
5292 C5          393         PUSH    BC
5293 F5          394         PUSH    AF
5294 F5          395         PUSH    AF      ; for ﾌﾗｸﾞ ﾚｼﾞｽﾀ
5295             396         ;
5295 CD E2 1F    397         CALL    #MPRINT
5298 0D          398         DB      $0D
5299 20 20 20    399         DM      "    SZ/H/PNC"
529C 53 5A 2F
529F 48 2F 50
52A2 4E 43
52A4 0D          400         DB      $0D
52A5 46 20 3D    401         DM      "F ="
52A8 00          402         DB      $00
52A9 E1          403         POP     HL      ; HL=AF
52AA CD 82 53    404         CALL    BIN$
52AD             405         ;
52AD CD E2 1F    406         CALL    #MPRINT
52B0 0D          407         DB      $0D
52B1 41 20 3D    408         DM      "A =$"
52B4 24
52B5 00          409         DB      $00
52B6 F1          410         POP     AF      ; AF=AF
52B7 CD C1 1F    411         CALL    #PRTHX
52BA             412         ;
52BA CD E2 1F    413         CALL    #MPRINT
52BD 20 20 42    414         DM      "  BC =$"
52C0 43 20 3D
52C3 24
52C4 00          415         DB      $00
52C5 E1          416         POP     HL      ; HL=BC
52C6 CD BE 1F    417         CALL    #PRTHL
52C9             418         ;
52C9 CD E2 1F    419         CALL    #MPRINT
52CC 20 20 44    420         DM      "  DE =$"
52CF 45 20 3D
52D2 24
52D3 00          421         DB      $00
52D4 E1          422         POP     HL      ; HL=DE
52D5 CD BE 1F    423         CALL    #PRTHL
52D8             424         ;
52D8 CD E2 1F    425         CALL    #MPRINT
52DB 20 20 48    426         DM      "  HL =$"
52DE 4C 20 3D
52E1 24
52E2 00          427         DB      $00
52E3 E1          428         POP     HL      ; HL=HL
52E4 CD BE 1F    429         CALL    #PRTHL
52E7             430         ;
52E7 CD E2 1F    431         CALL    #MPRINT
52EA 0D          432         DB      $0D
52EB 20 20 20    433         DM      "    SZ/H/PNC"
52EE 53 5A 2F
52F1 48 2F 50
52F4 4E 43
52F6 0D          434         DB      $0D
52F7 46 27 3D    435         DM      "F'="
52FA 00          436         DB      $00
52FB E1          437         POP     HL      ; HL=AF'
52FC CD 82 53    438         CALL    BIN$
52FF             439         ;
52FF CD E2 1F    440         CALL    #MPRINT
5302 0D          441         DB      $0D
5303 41 27 3D    442         DM      "A'=$"
5306 24
5307 00          443         DB      $00
5308 F1          444         POP     AF      ; AF=AF'
5309 CD C1 1F    445         CALL    #PRTHX
530C             446         ;
530C CD E2 1F    447         CALL    #MPRINT
530F 20 20 42    448         DM      "  BC'=$"
5312 43 27 3D
5315 24
5316 00          449         DB      $00
5317 E1          450         POP     HL      ; HL=BC'
5318 CD BE 1F    451         CALL    #PRTHL
531B             452         ;
531B CD E2 1F    453         CALL    #MPRINT
531E 20 20 44    454         DM      "  DE'=$"
5321 45 27 3D
5324 24
5325 00          455         DB      $00
5326 E1          456         POP     HL      ; HL=DE'
5327 CD BE 1F    457         CALL    #PRTHL
532A             458         ;
532A CD E2 1F    459         CALL    #MPRINT
532D 20 20 48    460         DM      "  HL'=$"
5330 4C 27 3D
5333 24
5334 00          461         DB      $00
5335 E1          462         POP     HL      ; HL=HL'
5336 CD BE 1F    463         CALL    #PRTHL
5339             464         ;
5339 CD EE 1F    465         CALL    #LTNL
533C CD E2 1F    466         CALL    #MPRINT
533F 49 58 20    467         DM      "IX =$"
5342 3D 24
```

```
5344 00          468         DB      $00
5345 E1          469         POP     HL      ; HL=IX
5346 CD BE 1F    470         CALL    #PRTHL
5349             471         ;
5349 CD E2 1F    472         CALL    #MPRINT
534C 20 20 49    473         DM      "  IY =$"
534F 59 20 3D
5352 24
5353 00          474         DB      $00
5354 E1          475         POP     HL      ; HL=IY
5355 CD BE 1F    476         CALL    #PRTHL
5358             477         ;
5358 CD EE 1F    478         CALL    #LTNL
535B CD E2 1F    479         CALL    #MPRINT
535E 53 50 20    480         DM      "SP =$"
5361 3D 24
5363 00          481         DB      $00
5364 21 02 00    482         LD      HL,2
5367 39          483         ADD     HL,SP   ; HL=SP
5368 CD BE 1F    484         CALL    #PRTHL
536B             485         ;
536B CD E2 1F    486         CALL    #MPRINT
536E 20 20 50    487         DM      "  PC =$"
5371 43 20 3D
5374 24
5375 00          488         DB      $00
5376 E1          489         POP     HL      ; HL=PC+1
5377 E5          490         PUSH    HL      ; Dummy
5378 2B          491         DEC     HL      ; HL=PC
5379 CD BE 1F    492         CALL    #PRTHL
537C CD EE 1F    493         CALL    #LTNL
537F             494         ;
537F C3 43 50    495         JP      HOT
5382             496 BIN$
5382 06 08       497         LD      B,8
5384             498 BIN2
5384 3E 18       499         LD      A,$30/2
5386 CB 25       500         SLA     L
5388 8F          501         ADC     A,A     ;$30+CY
5389 CD F4 1F    502         CALL    #PRINT
538C 10 F6       503         DJNZ    BIN2
538E C9          504         RET
538F             505 @BP
538F 00 00 00    506         DS      4*8+1   ; FLAG     = 1 Byte
5392 00 00 00
5395 00 00 00
5398 00 00 00
539B 00 00 00
539E 00 00 00
53A1 00 00 00
53A4 00 00 00
53A7 00 00 00
53AA 00 00 00
53AD 00 00 00
53B0             507                 ; ADDRESS = 2 Byte   *8
53B0             508                 ; DATA    = 1 Byte
53B0             509
53B0             510 DUMP
53B0 CD B2 1F    511         CALL    #HLHEX
53B3 30 03       512         JR      NC,DUMP1
53B5 2A FD 53    513         LD      HL,(@DADR)
53B8             514 DUMP1
53B8 E5          515         PUSH    HL
53B9 11 80 00    516         LD      DE,$0080
53BC 19          517         ADD     HL,DE
53BD 22 FD 53    518         LD      (@DADR),HL
53C0 E1          519         POP     HL
53C1             520         ;
53C1 CD 6B 51    521         CALL    PRN
53C4 06 10       522         LD      B,$10
53C6             523 DUMP2
53C6 CD BE 1F    524         CALL    #PRTHL
53C9 CD F1 1F    525         CALL    #PRINTS
53CC 0E 08       526         LD      C,8
53CE 11 ED 53    527         LD      DE,DUMP5
53D1             528 DUMP3
53D1 7E          529         LD      A,(HL)
53D2 CD C1 1F    530         CALL    #PRTHX
53D5 CD F1 1F    531         CALL    #PRINTS
53D8 ED A0       532         LDI
53DA 1B          533         DEC     DE
53DB 1A          534         LD      A,(DE)
53DC FE 20       535         CP      $20
53DE 30 03       536         JR      NC,DUMP4
53E0 3E 20       537         LD      A,$20
53E2 12          538         LD      (DE),A
53E3             539 DUMP4
53E3 13          540         INC     DE
53E4 03          541         INC     BC
53E5 0D          542         DEC     C
53E6 20 E9       543         JR      NZ,DUMP3
53E8             544         ;
53E8 CD E2 1F    545         CALL    #MPRINT
53EB 20 2F       546         DM      " /"
53ED             547 DUMP5
53ED 00 00 00    548         DS      8
53F0 00 00 00
53F3 00 00
53F5 0D 00       549         DB      $0D,$00
53F7             550         ;
53F7 10 CD       551         DJNZ    DUMP2
53F9 CD D6 1F    552         CALL    #LPTOF
53FC C9          553         RET
53FD             554 @DADR
53FD 00 00       555         DS      2
53FF             556
53FF             557 FIND    ; DUMMY
53FF             558 DISASM  ;   :
53FF             559 DVSET   ;   :
53FF             560 READ    ;   :
53FF             561 WRITE   ;   :
53FF C9          562         RET
5400             563
5400             564
```

清水和人流
プログラミング道場

(その4)

ピアニスト不屈のエディタ

Shimizu Kazuto

X68000をかついでドイツに向かった清水師範だが，実は出発前の混乱の最中に書き残していったと思われるプログラムと原稿が発見された。あのMMLリアルタイム入力にエディタを加えて威力倍増。アマグラマの心意気か。

さて，今回は半年ほど前にやったMMLリアルタイム入力に拡張を加え，MMLエディタに仕上げたので発表しよう。もともとあれはX68000のキーボードから音を実際のリズムで入力していき，MMLに変換しようというツールであった。この試み，アイディアはよいのだが，なかなかものにならない。それは次のような大問題を抱えているからである。

1) そんなにリズムを正確に弾き続けられる人がいない。やってみるとわかるけれども，かなりピアノのうまい人でも，聴いてみるとリズムが揃わないことはよくあるのだ。

2) 長いメロディだと音を間違えやすい。

3) 変則的リズム，たとえば3連符などを判断することができない。たとえできたとしても正確に入力できない。

4) メロディの音域が広すぎてキーボードの中では足りなくなってしまう。

これらの問題を正面から解決しようとすると非常に困難で，アマちゃんのプログラマにはちと荷が重い。てなわけで我がプログラミング道場では，できたデータをエディットできるようにしておけばよいと結論したのであった。そこで，リアルタイム入力ツールにエディタ機能を追加し「MMLエディタ」として仕上げてみたしだいである。

このエディタ機能によってもともとのツールの制約が一気に解消し，非常にフレキシブルなものとなる。たとえば，

1) できたMMLに音色，テンポなどの記述を挿入することで，変化に富んだ入力ができる。

2) 一度に複数のチェンジができるため，調性を変えたり，音をずらしたりすることが容易である。

3) 手で修正することにより，ミス入力の修正や複雑なリズムの記述ができる。

などがそれである。

また，もともとの入力方式は非常にわかりやすく，今までにないスピーディかつ容易なMMLの作成が可能である。

では，能書きはこれくらいにして，以下にコマンドとその機能を紹介しよう。1990年10月号の記事もあわせて参考にしてほしい。なお，プログラムはリアルタイム入力部分もあわせて一括掲載しよう。

メインメニューからの機能

プログラムを起動するとメインメニューが表示され，ここからの機能はコマンドを選択することから始まる。各コマンドは1文字なので入力は簡単である。各コマンドごとに説明しよう。

1) コマンド1～8：リアルタイム入力

数字の1～8はその番号のチャンネルにキーボードからリアルタイム入力によってデータを打ち込むモードを示す。入力後，拍子を示すBEEP音が始まるが，4拍分は見送り，5拍目から入力することになっている。キーボードに対する音の配置はKコマンドで見ることができる。休符はスペースキー，終了はリターンキーである。1拍に4つまで，1拍を4分音符とすると16分音符まで入力することが可能である。

2) コマンドC：Create MML

リアルタイム入力された生データ（これをエディットデータと呼ぶ）をMMLに変換する。変換は全パート同時に行われる。これを行わないと音楽を聞くことができない。

3) コマンドP：Play MML

コマンドCで変換してできたMMLを演奏する。Pを入力すると演奏するチャンネルを聞いてくるので数字を並べて入力する。数字の並びや区切り記号のあるなしには関係なく，存在する数字のチャンネルがすべて演奏される。また，0があると全チャンネルを演奏する。通常は0を入力して全パートを聞くか，MMLを作っていく段階では部分的なパートを聞くこともあるだろう。逆に，チャンネル番号をなにも指定しないと演奏をストップすることができる。

4) コマンドM：Mode change

入力モードを変更する。デフォルトはオーバーライト，すなわちそのパートの最初から上書きして，前のデータを残さない。Mコマンドを行うとモードが切り替わり，前のデータの最後につけ加えるモードになる。ただ，これはエディットデータに対してのみ有効である。Mコマンドを入力するたびにモードは変更される。

5) コマンドS：Save data

データをディスクにセーブする。セーブはカレントディレクトリに対して行われる。まずエディットデータとMMLのどちらをセーブするかを聞いてくるので，EかMで入力する。次にファイル名を聞いてくるので指定する。そのファイル名のファイルができセーブが実行される。

6) コマンドL：Load data

データをディスクからロードする。ファイル名を指定すればエディットデータかMMLかは自動的に判断されてロードされる。セーブ，ロードとも全パートを一度に扱う。

7) コマンドK：Show keyboard

キーボードの配置図を表示する。慣れないうちはこのコマンドによってどのキーがどの音かを確認できる。

8) コマンドU：Up octave

MMLに変換されたデータについて1オクターブ上に上げる。コマンドPと同様にチャンネルNo.を入力し，そのチャンネルに対して変換を行う。

9）コマンドD：Down octave

MMLに変換されたデータについて1オクターブ下に下げる。コマンドPと同様にチャンネルNo.を入力し，そのチャンネルに対して変換を行う。

10）コマンドO：change Onshoku

指定パートの音色を変える。音色は番号で指定する。音色と音色番号の関係についてはBASICマニュアルのm_trkの説明を参照のこと。ただし音色はMML変換後，Pコマンドを指定したときにのみ有効で，しかも各パートごとに1音色しか指定できない(すなわち曲の途中で音色は変えられない。やりたい場合はエディタを使用して行う)。

11）コマンドE：enter Editor

エディタを起動する。エディタ起動後の操作法は後述する。データを細かく変更するときに使用する。

12）コマンドQ：Quit system

MMLエディタを終了する。データをセーブしたらファイル名などをメモしておくようにね。お疲れさまでした。

エディタの機能

メインメニューでEコマンドを使用するとエディタが起動される。まずエディットデータとMMLのどちらをエディットするかを聞いてくるのでEかMで選択する。ここでEMまたはMEと入力すると両データを同時にエディットできる。次にパートを聞いてくるのでメインメニューのコマンドPと同様に入力する（全パート同時のエディットは0を入力する)。続いてエディタのコマンドを聞いてくるが，次にそれらのコマンドの内容を説明しよう。

1）コマンドC：Copy

エディタのコマンドは1行の中ですべての情報を入力してしまう。コマンドCの書式は，

C l_s-l_e,l_t,n

l_s ：スタート行番号

l_e ：エンド行番号

l_t ：ターゲット行番号

n ：コピー反復回数

となっている。この意味はl_s行目からl_e行目までの内容をl_t行にn回反復して挿入することを示す。たとえば，

C10－15,18,2

ならば10～15行を2回分，18行目から挿入することになる。また最後のnパラメータを省略すると1回分だけコピーを行う。

2）コマンドD：Delete

書式　Dl_s-l_e

l_s行目からl_e行目を削除してデータをつめる。l_sは省略すると1，l_eは省略すると最後の行を指す。

例）D5－7 → 5～7行を消す

D－8 → 1～8行を消す

D25－ → 25行以降を消す

3）コマンドF：Find

書式　Fl_s-l_e“文字列1”

l_s行からl_e行まで文字列1を探し，文字列1が含まれる行を表示する。

4）コマンドI：Insert

書式　I l_s

l_s行目に挿入する。挿入する内容はコマンド入力後に聞いてくるl_s行から最終行までが1行ずつずれて1行長くなる。

5）コマンドH：Help

エディタコマンドの一覧を表示する。各コマンドの入力書式も表示される。

6）コマンドL：List

書式　Ll_s-l_e

l_s行目からl_e行目のリストを表示する。l_sを省略すると1行目から，l_eを省略すると最後の行までを表示する。

7）コマンドP：Part

エディットの対象となるパートすなわちチャンネルを変更する。メインメニューのコマンドP（Play）と同様にパートを入力すると，各コマンドの対象が指定されたパートになる。

8）コマンドO：Overwrite

書式：Ol_s

l_s行に上書きする。コマンドを打ち込むと上書きする文字列を聞いてくる。Iコマンドとの使い分けが必要である。

9）コマンドQ：Quit

エディット作業を終了する。メインメニューに戻る。

10）コマンドS：Substitute

書式：Sl_s-l_e”文字列1””文字列2”

l_s行からl_e行までの範囲のすべての文字列1を文字列2に変更する。Fコマンドと併用することによって特定の音を変えたりと応用範囲が広い。

さらなる発展を望む

以上，リアルタイム入力にエディタの機能がついてかなり応用範囲が広がった。同じようなパターンのところはCopyやSubstituteコマンドでどんどん作っていくことができる。これは私自身が使いやすいように作ったプログラムだが，BASICで書かれているので，各自の考えで独自のコマンドを追加して使ってほしい。

注）この記事は清水氏がドイツに出発する前に書かれた最後の原稿です。帰国後の新たな展開を期待しましょう。

●リスト1

```
   10 /*********************************
   20 /* program for creating MML      *
   30 /*     1990.8.22 by k.shimizu    *
   40 /*********************************
   50 /*********************************
   60 /* data initializing             *
   70 /*********************************
   80 dim str c$(100,8)[4],m$(100,8)[50],mwork$[50],mwk$[50]
   90 dim int d(100,8),jmax(8),ipart(8),e(100,8)
  100 int j=1,f=0,i=0,k=0,l=1,iold=0,iimax=0,ll=100,ih=4,r,r0,
      r1,irflag=0,lm=16,dum=0,imode,npart=8,ifkey=0,j1,r2,
      r3,i1,jj,i2
  110 str o$(8)[4]={"@1","@1","@1","@1","@1","@1","@1","@1","@1"
}
  120 str ba$[47]=
      "zxcvbnm,./qwertyuiop@[asdfghjkl;:1234567890-^¥ "
  130 str ca$(48)[4]={"","3c","3d","3e","3f","3g","3a","3b","4c"
,"4d","4e","4f","4g","4a","4b","5c","5d","5e","5f","5g","5a","5b
","6c","2b","3c+","3d+","3e+","3f+","3g+","3a+","3b+","4c+","4d+
","4e+","4e+","4f+","4g+","4a+","4b+","5c+","5d+","5e+"}
  140 str cb$(47)[3]={"","ﾄﾞ","ﾚ","ﾐ","ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ﾚ",
"ﾐ","ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ﾚ","ﾐ","ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ｼ","ﾄﾞ#
","ﾚ#","ﾌｧ","ﾌｧ#","ｿ#","ﾗ#","ﾄﾞ","ﾄﾞ#","ﾚ#","ﾌｧ","ﾌｧ","ﾌｧ#","ｿ#"
,"ﾗ#","ﾄﾞ","ﾄﾞ#","ﾚ#","ﾌｧ","ﾏｧ#","ｿ#","ﾗ#","ﾄﾞ","ﾄﾞ#","."}
  150 str cc$(47)[3]={"","ﾄﾞ","ﾚ","ﾐ","ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ﾚ",
"ﾐ","ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ﾚ","ﾐ","ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ｼ","ﾚb"
,"ﾐb","ﾌｧ","ｿb","ﾗb","ｼb","ﾄﾞ","ﾚb","ﾐb","ﾌｧ","ﾌｧ","ｿb","ﾗb","ｼb
","ﾄﾞ","ﾚb","ﾐb","ﾌｧ","ｿb","ﾗb","ｼb","ﾄﾞ","ﾚb","."}
  160 ca$(42)="5f+":ca$(43)="5g+":ca$(44)="5a+":
      ca$(45)="5b+":ca$(46)="6c+":ca$(47)="5r"
  170 str e$(8)[4],b$="",a$="",cl$="",cm$="",fil$,part$,dum$,mod
e$(2)[10]={"over write","merge"},crcl,com$,dum1$,dm$
  180 crcl=chr$(13)+chr$(10)
  190 for i=1 to 47:ca$(i)="o"+ca$(i):next
  200 /*********************************
  210 /* initialize FM                 *
  220 /*********************************
  230 m_init()
```

```
240 for i=1 to 8:m_alloc(i,8000):next
250 for i=1 to 8:m_assign(i,i):next
260 for i=1 to 8:jmax(i)=0:next
270 /*********************************
280 /* main loop (ended by "e")      *
290 /*********************************
300 repeat
310    if ifkey=0 then {
320      menu()
330      print:print "*** now in the [";mode$(imode);"] mode"
340      print "*** please start at the 5th beep":print
350    } else {
360       ifkey=0
370    }
380    cl$=""
390    input "input command";cl$
400    cl$=strupr(cl$)
410    l=asc(cl$)-48
420 /*********************************
430 /* input sound by keyboard      *
440 /*********************************
450    if l>=1 and l<=8 then {
460       if imode=1 then j=jmax(l)+1:i=d(jmax(l),l)-l1
470       repeat
480       jmax(l)=100
490       repeat
500          a$=inkey$(0)
510          i=i+1
520          if i mod l1/4=0 then beep
530          if i=l1+l1/4+l1/lm/2-1 and j=1 then a$=" "
540          if a$="" then for k=1 to 200:next
550       until a$<>""
560       if a$=chr$(13) then jmax(l)=j:d(j,l)=i else {
570          d(j,l)=i:print ".";
580          m_init()
590          f=instr(1,ba$,a$)
600          m_trk(l,o$(l)+"l1"+ca$(f))
610 /*       m_trk(l,"&l1"+ca$(f))
620 /*       m_trk(l,"&l1"+ca$(f))
630 /*       m_trk(l,"&l1"+ca$(f))
640 /*       m_trk(l,"&l1"+ca$(f))
650 /*       m_trk(l,"&l1"+ca$(f))
660          c$(j,l)=ca$(f)
670          m_play()
680          j=j+1
690       }
700       until j=jmax(l)
710       jmax(l)=jmax(l)-1:j=1:i=0
720       print
730    }
740 /*********************************
750 /* create MML from array d,c$   *
760 /*********************************
770    if cl$="C" then {
780       print "now creating MML"
790       for l=1 to 8
800       for j=1 to jmax(l)+1
810          e(j,l)=int((d(j,l)+l1/lm/2)*lm/l1)
820       next
830       next
840       for l=1 to 8
850          print ".";
860       for j=1 to jmax(l)
870          r=e(j+1,l)-e(j,l)
880          mwork$=""
890          r1=r/(4*4)
900          r=r -r1*4*4
910          if r=0 and r1>=1 then r=16:r1=r1-1
920          irflag=0:mwk$=c$(j,l)
930  if instr(1,c$(j,l),"r")<>0 then c$(j,l)="":irflag=1
940  if r= 1 then m$(j,l)="l16"+c$(j,l)
950  if r= 2 then m$(j,l)="l8" +c$(j,l)
960  if r= 3 then m$(j,l)="l8"+c$(j,l)+"."
970  if r= 4 then m$(j,l)="l4"+c$(j,l)
980  if r= 5 then m$(j,l)="l4"+c$(j,l)+"&l16"+c$(j,l)
990  if r= 6 then m$(j,l)="l4"+c$(j,l)+"."
1000 if r= 7 then m$(j,l)="l4"+c$(j,l)+".."
1010 if r= 8 then m$(j,l)="l2"+c$(j,l)
1020 if r= 9 then m$(j,l)="l2"+c$(j,l)+"&l16"+c$(j,l)
1030 if r=10 then m$(j,l)="l2"+c$(j,l)+"&l8"+c$(j,l)
1040 if r=11 then m$(j,l)="l2"+c$(j,l)+"&l8"+c$(j,l)+"."
1050 if r=12 then m$(j,l)="l2"+c$(j,l)+"."
1060 if r=13 then m$(j,l)="l2"+c$(j,l)+".l16"+c$(j,l)
1070 if r=14 then m$(j,l)="l2"+c$(j,l)+".&l8"+c$(j,l)
1080 if r=15 then m$(j,l)="l2"+c$(j,l)+".&l8"+c$(j,l)+"."
1090 if r=16 then m$(j,l)="l1"+c$(j,l)
1100       mwork$="l1"+c$(j,l)
1110       if r<>0 then mwork$=mwork$+"&"
1120       c$(j,l)=mwk$
1130       for k=1 to r1:m$(j,l)=mwork$+m$(j,l):next
1140       mwork$=""
1150       for i=1 to 20
1160          if irflag=1 then {
1170          if mid$(m$(j,l),i,1)="l" then mwork$=mwork$+"r"
1180          if mid$(m$(j,l),i,l)<>"l" then
                 mwork$=mwork$+mid$(m$(j,l),i,l)
1190          }
1200       next
1210       if irflag=1 then m$(j,l)=mwork$
1220       next
1230       next
1240       print
1250    }
1260 /*********************************
1270 /* play MML                     *
1280 /*********************************
1290    if cl$="P" then {
1300       input "input part number(0 for all part)";part$
1310       npart=0:for i=1 to 8:ipart(i)=0:next
1320       for i=1 to 8
1330          if (instr(1,part$,itoa(i))<>0 or
                 instr(1,part$,"0")     <>0 ) then {
1340             npart=npart+1
1350             ipart(npart)=i
1360          }
1370       next
1380       print "now playing !"
1390       m_init()
1400       for l=1 to 8
1410        m_trk(l,o$(ipart(l)))
1420        for j=1 to jmax(ipart(l))
1430          m_trk(l,m$(j,ipart(l)))
1440       next
1450       next
1460       m_play()
1470       j=1:i=0:l=0
1480    }
1490 /*********************************
1500 /* mode change                  *
1510 /*********************************
1520    if cl$="M" then {
1530       print "mode was changed !"
1540       imode=1-imode
1550    }
1560 /*********************************
1570 /* save data                    *
1580 /*********************************
1590    if cl$="S" then {
1600       input "choose m:MML or e:EDIT DATA";cm$
1610       input "input file name";fil$
1620       input "input part number(0 for all part)";part$
1630       npart=0:for i=1 to 8:ipart(i)=0:next
1640       for i=1 to 8
1650          if (instr(1,part$,itoa(i))<>0 or
                 instr(1,part$,"0")     <>0 ) then {
1660             npart=npart+1
1670             ipart(npart)=i
1680          }
1690       next
1700       cm$=left$(strupr(cm$),1)
1710       ifil=fopen(fil$,"c")
1720       fwrites(cm$+crcl,ifil)
1730       fwrites(itoa(npart)+crcl,ifil)
1740       print "now saving";
1750       for i=1 to npart
1760          print ".";
1770          fwrites(itoa(ipart(i))+crcl,ifil)
1780          fwrites(itoa(jmax(ipart(i)))+crcl,ifil)
1790          fwrites(o$(ipart(i))+crcl,ifil)
1800          for j=1 to jmax(ipart(i))+1
1810             if cm$="M" then
                    fwrites(m$(j,ipart(i))+crcl,ifil)
1820             if cm$="E" then
                    fwrites(c$(j,ipart(i))+crcl,ifil)
1830             if cm$="E" then
                    fwrites(itoa(d(j,ipart(i)))+crcl,ifil)
1840          next
1850       next
1860       print
1870       fclose(ifil)
1880       if cm$="M" then print "MML is saved in ";fil$
1890       if cm$="E" then print "EDIT DATA is saved in ";fil$
1900    }
1910 /*********************************
1920 /* load data                    *
1930 /*********************************
1940    if cl$="L" then {
1950       input "input file name";fil$
1960       ifil=fopen(fil$,"r")
1970       freads(cm$,ifil)
1980       cm$=strupr(cm$)
1990       freads(dum$,ifil)
2000       npart=atoi(dum$)
2010       print "now loading";
2020       for i=1 to npart
2030          print ".";
2040          freads(dum$,ifil)
2050          ipart(i)=atoi(dum$)
2060          freads(dum$,ifil)
2070          r=jmax(ipart(i))
2080          if imode=0 then r=0
2090          r1=d(r,ipart(i))+1
2100          jmax(ipart(i))=atoi(dum$)
2110          freads(dum$,ifil)
2120          o$(ipart(i))=dum$
2130          for j=r+1 to r+jmax(ipart(i))+1
2140             if cm$="M" then freads(dum$,ifil)
2150             if cm$="M" then m$(j,ipart(i))=dum$
2160             if cm$="E" then freads(dum$,ifil)
2170             if cm$="E" then c$(j,ipart(i))=dum$
2180             if cm$="E" then freads(dum$,ifil)
2190             if j=r+1 and r>0 then r2=atoi(dum$)
2200             if cm$="E" then d(j,ipart(i))=atoi(dum$)
2210             if cm$="E" and r>0 then
                    d(j,ipart(i))=r1+atoi(dum$)-r2
2220          next
2230       next
2240       print
2250       fclose(ifil)
2260       if cm$="M" then print "MML is loaded from ";fil$
2270       if cm$="E" then print "EDIT DATA is loaded from "
                          ;fil$
2280    }
2290 /*********************************
2300 /* display keyboard             *
2310 /*********************************
2320    if cl$="K" then {
2330       keyboard()
2340    }
2350 /*********************************
2360 /* up octave                    *
2370 /*********************************
2380    if cl$="U" then {
2390       input "input part number(0 for all part)";part$
```

```
2400       npart=0:for i=1 to 8:ipart(i)=0:next
2410       for i=1 to 8
2420          if (instr(1,part$,itoa(i))<>0 or
                instr(1,part$,"0")     <>0 ) then {
2430             npart=npart+1
2440             ipart(npart)=i
2450             for j=1 to jmax(ipart(npart))
2460                r=instr(1,c$(j,ipart(npart)),"o")
2470                r0=asc(mid$(c$(j,ipart(npart)),r+1,1))
2480                if r0>=asc("0") and r0<=asc("7") then {
2490                   r1=strlen(c$(j,ipart(npart)))-r-1
2500                   dum$=left$(c$(j,ipart(npart)),r)+
                            chr$(r0+1)+
                            right$(c$(j,ipart(npart)),r1)
2510                   c$(j,ipart(npart))=dum$
2520                }
2530             next
2540          }
2550       next
2560    }
2570 /*********************************
2580 /* down octave                    *
2590 /*********************************
2600    if cl$="D" then {
2610       input "input part number(0 for all part)";part$
2620       npart=0:for i=1 to 8:ipart(i)=0:next
2630       for i=1 to 8
2640          if (instr(1,part$,itoa(i))<>0 or
                instr(1,part$,"0")     <>0 ) then {
2650             npart=npart+1
2660             ipart(npart)=i
2670             for j=1 to jmax(ipart(npart))
2680                r=instr(1,c$(j,ipart(npart)),"o")
2690                r0=asc(mid$(c$(j,ipart(npart)),r+1,1))
2700                if r0>=asc("1") and r0<=asc("8") then {
2710                   r1=strlen(c$(j,ipart(npart)))-r-1
2720                   dum$=left$(c$(j,ipart(npart)),r)+
                            chr$(r0-1)+
                            right$(c$(j,ipart(npart)),r1)
2730                   c$(j,ipart(npart))=dum$
2740                }
2750             next
2760          }
2770       next
2780    }
2790 /*********************************
2800 /* change onshoku                 *
2810 /*********************************
2820    if cl$="O" then {
2830       input "input part number(0 for all part)";part$
2840       input "input onshoku number(1-200)";r
2850       npart=0:for i=1 to 8:ipart(i)=0:next
2860       for i=1 to 8
2870          if (instr(1,part$,itoa(i))<>0 or
                instr(1,part$,"0")     <>0 ) then {
2880             npart=npart+1
2890             ipart(npart)=i
2900             o$(ipart(npart))="@"+itoa(r)
2910          }
2920       next
2930    }
2940 /*********************************
2950 /* edit MML or EDIT DATA          *
2960 /*********************************
2970    if cl$="E" then {
2980       input "choose m:MML or e:EDIT DATA";cm$
2990       cm$=strupr(cm$)
3000       npart=1
3010       ipart(npart)=1
3020       repeat
3030          linput "EDITOR>>";com$
3040          com$=strupr(com$)
3050 /* C:copy command
3060          if left$(com$,1)="C" then {
3070             r=instr(1,com$,"-")
3080             r2=instr(1,com$,",")
3090             r3=instr(r2+1,com$,",")
3100             i=strlen(com$)
3110             if r>=3 then {
3120                r0=atoi(mid$(com$,2,r-2))
3130             } else {
3140                r0=1
3150             if i>1 and r=0 then r0=atoi(mid$(com$,2,i-1))
3160             }
3170             if r<r2 and r>0 then {
3180                r1=atoi(mid$(com$,r+1,r2-r-1))
3190             } else {
3200                r1=0
3210                for j=1 to npart
3220                if r1<jmax(ipart(j)) then r1=jmax(ipart(j))
3230                next
3240                if r=0 then r1=r0
3250             }
3260             if r3>0 then {
3270                k=atoi(mid$(com$,r2+1,r3-r2-1))
3280                kk=1
3290                if r3<i then kk=atoi(mid$(com$,r3+1,i-r3))
3300             } else {
3310                k=atoi(mid$(com$,r2+1,i-r2))
3320                kk=1
3330             }
3340             for j=1 to npart
3350                jj=ipart(j)
3360                for i1=k to jmax(jj)
3370                   i=k+jmax(jj)-i1
3380                   if instr(1,cm$,"E")<>0 then {
3390                      c$(i+kk*(r1-r0+1),jj)=c$(i,jj)
3400                      d(i+kk*(r1-r0+1),jj)=d(i,jj)
3410                   }
3420                   if instr(1,cm$,"M")<>0 then {
3430                      m$(i+kk*(r1-r0+1),jj)=m$(i,jj)
3440                   }
3450                next
3460             next
3470             for j=1 to npart
3480                jj=ipart(j)
3490                for i=1 to kk
3500                for i1=r0 to r1
3510                 if instr(1,cm$,"E")<>0 then {
3520                 if i1<k then {
3530                  c$(k+(i-1)*(r1-r0+1)+i1-r0,jj)=c$(i1,jj)
3540                  d(k+(i-1)*(r1-r0+1)+i1-r0,jj)=d(i1,jj)
3550                 }
3560                 if i1>=k then {
3570                  c$(k+(i-1)*(r1-r0+1)+i1-r0,jj)=
                         c$(i1+kk*(r1-r0+1),jj)
3580                  d(k+(i-1)*(r1-r0+1)+i1-r0,jj)=
                         d(i1+kk*(r1-r0+1),jj)
3590                 }
3600                 }
3610                 if instr(1,cm$,"M")<>0 then {
3620                 if i1<k then {
3630                  m$(k+(i-1)*(r1-r0+1)+i1-r0,jj)=m$(i1,jj)
3640                 }
3650                 if i1>=k then {
3660                  m$(k+(i-1)*(r1-r0+1)+i1-r0,jj)=
                         m$(i1+kk*(r1-r0+1),jj)
3670                 }
3680                 }
3690                next
3700                next
3710                jmax(jj)=jmax(jj)+kk*(r1-r0+1)
3720             next
3730          }
3740 /* D:delete command
3750          if left$(com$,1)="D" then {
3760             r=instr(1,com$,"-")
3770             i=strlen(com$)
3780             if r>=3 then {
3790                r0=atoi(mid$(com$,2,r-2))
3800             } else {
3810                r0=1
3820             if i>1 and r=0 then r0=atoi(mid$(com$,2,i-1))
3830             }
3840             if r<i and r>0 then {
3850                r1=atoi(mid$(com$,r+1,i-r))
3860             } else {
3870                r1=0
3880                for j=1 to npart
3890                if r1<jmax(ipart(j)) then r1=jmax(ipart(j))
3900                next
3910                if r=0 then r1=r0
3920             }
3930             for j=1 to npart
3940                for i=r0 to jmax(ipart(j))-r1+r0
3950                   j1=ipart(j)
3960                   if instr(1,cm$,"E")<>0 then {
3970                      c$(i,j1)=c$(i+r1-r0+1,j1)
3980                      d(i,j1)=d(i+r1-r0+1,j1)
3990                   }
4000                   if instr(1,cm$,"M")<>0 then {
4010                      m$(i,j1)=m$(i+r1-r0+1,j1)
4020                   }
4030                next
4040                jmax(j1)=jmax(j1)-r1+r0-1
4050             next
4060          }
4070 /* F:find command
4080          if left$(com$,1)="F" then {
4090             r=instr(1,com$,"-")
4100             i=strlen(com$)
4110             if r>=3 then {
4120                r0=atoi(mid$(com$,2,r-2))
4130             } else {
4140                r0=1
4150             if i>1 and r=0 then r0=atoi(mid$(com$,2,i-1))
4160             }
4170             if r<i and r>0 then {
4180                r1=atoi(mid$(com$,r+1,i-r))
4190             } else {
4200                r1=0
4210                for j=1 to npart
4220                if r1<jmax(ipart(j)) then r1=jmax(ipart(j))
4230                next
4240                if r=0 then r1=r0
4250             }
4260             r2=instr(1,com$,chr$(34))
4270             r3=instr(r2+1,com$,chr$(34))
4280             if r2>1 and r3>r2 then {
4290             dum$=mid$(com$,r2+1,r3-1-r2)
4300             for i=r0 to r1
4310             for j=1 to npart
4320                j1=ipart(j)
4330                if instr(1,cm$,"E")<>0 then {
4340                   if instr(1,c$(r0,j1),dum$)<>0 then {
4350                      print "c$(";r0;",";j1;")=";c$(r0,j1)
4360                   }
4370                 if instr(1,itoa(d(r0,j1)),dum$)<>0 then {
4380                      print "d(";r0;",";j1;")=";d(r0,j1)
4390                   }
4400                }
4410                if instr(1,cm$,"M")<>0 then {
4420                   if instr(1,m$(r0,j1),dum$)<>0 then {
4430                      print "m$(";r0;",";j1;")=";m$(r0,j1)
4440                   }
4450                }
4460             next
4470             next
4480             }
4490          }
4500 /* I:insert command
4510          if left$(com$,1)="I" then {
4520             i=strlen(com$)
4530             if i=1 then r0=r0+1
```


▶たまに字を書くと，“そ”より“SO”と頭に浮かんでくる。病気かな？
宮迫　啓彰(16)鹿児島県

```
4540            if i>1 then r0=atoi(right$(com$,i-1))
4550            for j=1 to npart
4560               j1=ipart(j)
4570               for i1=r0 to jmax(j1)
4580                  i=r0+jmax(j1)-i1
4590                  if instr(1,cm$,"E")<>0 then {
4600                     c$(i,j1)=c$(i-1,j1)
4610                     d(i,j1)=d(i-1,j1)
4620                  }
4630                  if instr(1,cm$,"M")<>0 then {
4640                     m$(i,j1)=m$(i-1,j1)
4650                  }
4660               next
4670               if instr(1,cm$,"E")<>0 then {
4680                  print "c$(";r0;",";j1;")=";c$(r0+1,j1)
4690                  input "input c$";dum$
4700                  if dum$<>"" then c$(r0,j1)=dum$
4710                  print "d(";r0;",";j1;")=";d(r0+1,j1)
4720                  input "input d";dum$
4730                  if dum$<>"" then d(r0,j1)=atoi(dum$)
4740               }
4750               if instr(1,cm$,"M")<>0 then {
4760                  print "m$(";r0;",";j1;")=";m$(r0+1,j1)
4770                  input "input m$";dum$
4780                  if dum$<>"" then m$(r0,j1)=dum$
4790               }
4800               jmax(j1)=jmax(j1)+1
4810            next
4820         }
4830 /* L:list command
4840         if left$(com$,1)="L" then {
4850            r=instr(1,com$,"-")
4860            i=strlen(com$)
4870            if r>=3 then {
4880               r0=atoi(mid$(com$,2,r-2))
4890            } else {
4900               r0=1
4910            if i>1 and r=0 then r0=atoi(mid$(com$,2,i-1))
4920            }
4930            if r<i and r>0 then {
4940               r1=atoi(mid$(com$,r+1,i-r))-1
4950            } else {
4960               r1=0
4970               for j=1 to npart
4980               if r1<jmax(ipart(j)) then r1=jmax(ipart(j))
4990               next
5000               if r=0 and i>1 then r1=r0-1
5010            }
5020            for i=r0 to r1+1
5030               for j1=1 to npart
5040                  j=ipart(j1)
5050                  dum$=inkey$(0)
5060                  if dum$<>"" then i=r1+1 else {
5070                  if instr(1,cm$,"E")<>0 then {
5080                     print i,j,c$(i,j),d(i,j)
5090                  }
5100                  if instr(1,cm$,"M")<>0 then {
5110                     print i,j,m$(i,j)
5120                  }
5130                  }
5140               next
5150            next
5160         }
5170 /* P:part command
5180       if left$(com$,1)="P" then {
5190          part$=mid$(com$,2,strlen(com$)-1)
5200          if strlen(com$)>1 then {
5210             npart=0:for i=1 to 8:ipart(i)=0:next
5220          }
5230          for i=1 to 8
5240             if (instr(1,part$,itoa(i))<>0 or
                     instr(1,part$,"0")    <>0 ) then {
5250                npart=npart+1
5260                ipart(npart)=i
5270                print "***part";i;"***";jmax(i)
5280             }
5290          next
5300       }
5310 /* O:overwrite command
5320         if left$(com$,1)="O" then {
5330            i=strlen(com$)
5340            if i=1 then r0=r0+1
5350            if i>1 then r0=atoi(right$(com$,i-1))
5360            for j=1 to npart
5370               j1=ipart(j)
5380               if instr(1,cm$,"E")<>0 then {
5390                  print "c$(";r0;",";j1;")=";c$(r0,j1)
5400                  input "input c$";dum$
5410                  if dum$<>"" then c$(r0,j1)=dum$
5420                  print "d(";r0;",";j1;")=";d(r0,j1)
5430                  input "input d";dum$
5440                  if dum$<>"" then d(r0,j1)=atoi(dum$)
5450               }
5460               if instr(1,cm$,"M")<>0 then {
5470                  print "m$(";r0;",";j1;")=";m$(r0,j1)
5480                  input "input m$";dum$
5490                  if dum$<>"" then m$(r0,j1)=dum$
5500               }
5510            next
5520         }
5530 /* S:substitute command
5540         if left$(com$,1)="S" then {
5550            r=instr(1,com$,"-")
5560            i=strlen(com$)
5570            if r>=3 then {
5580               r0=atoi(mid$(com$,2,r-2))
5590            } else {
5600               r0=1
5610            if i>1 and r=0 then r0=atoi(mid$(com$,2,i-1))
5620            }
5630            if r<i and r>0 then {
5640               r1=atoi(mid$(com$,r+1,i-r))
5650            } else {
5660               r1=0
5670               for j=1 to npart
5680               if r1<jmax(ipart(j)) then r1=jmax(ipart(j))
5690               next
5700               if r=0 then r1=r0
5710            }
5720            r2=instr(1,com$,chr$(34))
5730            r3=instr(r2+1,com$,chr$(34))
5740            if r2>1 and r3>r2 then {
5750            dum$=mid$(com$,r2+1,r3-1-r2)
5760            r2=instr(r3+1,com$,chr$(34))
5770            r3=instr(r2+1,com$,chr$(34))
5780            if r2>1 and r3>r2 then {
5790            dum1$=mid$(com$,r2+1,r3-1-r2)
5800            i1=strlen(dum$)
5810            for i=r0 to r1
5820            for j=1 to npart
5830               j1=ipart(j)
5840               if instr(1,cm$,"E")<>0 then {
5850                  i2=instr(1,c$(r0,j1),dum$)
5860                  if i2<>0 then {
5870                     print "c$(";r0;",";j1;")=";c$(r0,j1)
5880                     dm$=c$(r0,j1)
5890                  c$(r0,j1)=left$(dm$,i2-1)+dum1$+
                           mid$(dm$,i1+i2,strlen(dm$))
5900                     print "  =>  ";c$(r0,j1)
5910                  }
5920               }
5930               if instr(1,cm$,"M")<>0 then {
5940                  i2=instr(1,m$(r0,j1),dum$)
5950                  if i2<>0 then {
5960                     print "m$(";r0;",";j1;")=";m$(r0,j1)
5970                     dm$=m$(r0,j1)
5980                  m$(r0,j1)=left$(dm$,i2-1)+dum1$+
                           mid$(dm$,i1+i2,strlen(dm$))
5990                     print "  =>  ";m$(r0,j1)
6000                  }
6010               }
6020            next
6030            next
6040            }
6050            }
6060         }
6070    until left$(com$,1)="Q"
6080 }
6090 if cl$="" then m_init()
6100 until cl$="Q"
6110 end
6120 /**********************************
6130 /* function menu is only for     *
6140 /*      drawing main menu        *
6150 /**********************************
6160 func menu()
6170    print "************************"
6180    print "* 1-8:sound input      *"
6190    print "*  c:create MML        *"
6200    print "*  p:play   MML        *"
6210    print "*  m:mode   change     *"
6220    print "*  s:save   data       *"
6230    print "*  l:load   data       *"
6240    print "*  k:show   keyboard   *"
6250    print "*  u:up     octave     *"
6260    print "*  d:down   octave     *"
6270    print "*  o:change onshoku    *"
6280    print "*  e:enter  editor     *"
6290    print "*  q:quit   system     *"
6300    print "************************"
6310    return()
6320 endfunc
6330 /**********************************
6340 /* function keyboard is for      *
6350 /*      drawing key position     *
6360 /**********************************
6370 func keyboard()
6380    print "*******************************************"
6390    print "* ﾌｧﾌｧ# ｿ# ﾗ# ﾄﾞﾄﾞ# ﾚ# ﾌｧﾌｧ# ｿ# ﾗ# ﾄﾞﾄﾞ# *"
6400    print "* 1  2  3  4  5  6  7  8  9  0  -  ^  ¥  *"
6410    print "*                                        *"
6420    print "*  ﾌｧ ｿ  ﾗ  ｼ  ﾄﾞ ﾚ  ﾐ  ﾌｧ ｿ  ﾗ  ｼ  ﾄﾞ    *"
6430    print "*  q  w  e  r  t  y  u  i  o  p  @  [    *"
6440    print "*                                        *"
6450    print "*   ｼ ﾄﾞ# ﾚ# ﾌｧﾌｧ# ｿ# ﾗ# ﾄﾞﾄﾞ# ﾚ# ﾌｧ      *"
6460    print "*   a  s  d  f  g  h  j  k  l  ;  :      *"
6470    print "*                                        *"
6480    print "*    ﾄﾞ ﾚ  ﾐ  ﾌｧ ｿ  ﾗ  ｼ  ﾄﾞ ﾚ  ﾐ         *"
6490    print "*    z  x  c  v  b  n  m  ,  .  /        *"
6500    print "*******************************************"
6510    ifkey=1
6520    return()
6530 endfunc
6540 /**********************************
6550 /* the end of the program        *
6560 /* (lines below are for debug)   *
6570 /**********************************
6580 for i=1 to jmax(1)
6590    print d(i+1,ii)-d(i,ii);c$(i,ii);" ";m$(i,ii)
6600 next
6610 end
6620 str dum$,fil$
6630 input "input file name ";fil$
6640 ifil=fopen(fil$,"r")
6650 repeat
6660 nfil=freads(dum$,ifil)
6670 print nfil,"'";dum$;"'"
6680 until 1=0
```

▶雪の降る日に車を運転していて，フロントガラスに当たって左右に別れる雪を見ていたら，気分が悪くなった。雪の日は事故が多いのに納得。追伸，さすがに同乗者には気分が悪いとはいえなかった。　川並　仁(19)三重県

愛読者プレゼント

❶ 中華大仙

シャープ ☎03(3260)1161

X68000用 5"2HD版 7,900円(税別) 3名

アーケードから忠実に移植された横スクロールシューティングゲーム。比較的やさしいので，シューティングは苦手という人でも楽しめるはず。

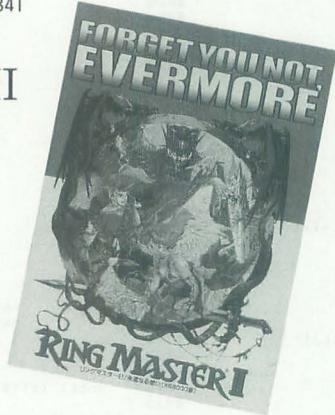
❷ リングマスターII

ホビージャパン ☎03(3354)9341

X68000用 5"2HD版3枚組

9,800円(税別) 2名

テーブルトークの老舗，ホビージャパンが放つ本格RPG「リングマスター」の第2作。前作から1年，待ち望んでいた人も多いことでしょう。

❸ エメラルドドラゴン

グローディア ☎03(3220)5226

X68000用
5"2HD版6枚組

9,800円(税別)

2名

PC-8801版で発売され根強いファンを持つRPG。X68000への移植にあたって，新たにグラフィックも描き起こされ，ジョイスティックやマウスにも対応するようになった。

❹ SOUND WORLD OF POPULOUS

ポニーキャニオン ☎03(3221)3161

2,500円(税込) 2名

いわずと知れたポピュラスのCDが発売。プロミストランド，スーファミオリジナルのワールドタイトルなどが収録されている。またテーマ曲は葛生千夏が唄っている。

❺ サイトロンビデオゲームミュージック年鑑1990

ポニーキャニオン ☎03(3221)3161

4,500円(税込み) 2名

月刊ゲーメストのゲーメスト大賞VGM部門で選ばれた1990年のビデオゲーム10作品をCD化。アレンジバージョンとオリジナルバージョン（全曲）を収録という豪華版だ。

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1991年4月18日の到着分までとします。当選者の発表は1991年6月号で行います。

2月号プレゼント当選者

1イメージファイト（神奈川県）志塚佳彦（愛知県）島倉功（滋賀県）吉田尚史 2銀河英雄伝説II（北海道）太田哲雄（大阪府）池田篤史（広島県）田頭修 3プロミストランド（千葉県）須佐英之（群馬県）備後秀明（香川県）三島武典 4大戦略カレンダー（東京都）中村幸司 山本修（大阪府）越田剛史（福岡県）西園誠（熊本県）鍋田崇 5テレホンカード（青森県）米田充（東京都）庄島賢一（埼玉県）谷島孝行 （敬称略）

以上の方々が当選されました。おめでとうございます。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。また，公正取引委員会の告示により，このプレゼントに当選された方は，この号の他の懸賞には当選できない場合がありますのでご了承ください。

◆X-OVER NIGHT (クロスオーバーナイト)

[第11話]

アメリカ人の気質

TAKAHARA HIDEKI 高原 秀己

1989年12月，全日本プロレス「世界最強タッグリーグ戦」での出来事。天龍源一郎と日米合体タッグを組んでスタン・ハンセンがリーグ戦に出場した。ハンセンが天龍を盛んに持ち上げたこともあって，なかなか息が合ったコンビネーションを見せて快進撃が続いた。リーグ戦の行方は最終戦までもつれ込み、天龍がテリー・ゴディ，スティーブ・ウイリアムス組にギブアップをとられて優勝を逃した。

ここで，ハンセンの態度が変わった。苦労して戦ってきたのに，最後になってギブアップを口にするとは，といわんばかりの形相で，天龍を殴る蹴る，ロープで絞めあげるの大暴れ。その後，天龍が全日本プロレスを退職した翌年4月まで，ハンセンの「制裁」は続いた。

もともと農耕民族のわれわれ日本人とは違い，狩猟民族の喜怒哀楽の激しさは当然のこととも受け取れる。しかし，それ以上のアメリカ人の特徴ともいえる，独特の身勝手さを象徴的に体現したケースとして，この全日本プロレスでの出来事は印象に強く残っている。

自分の規定する枠組みの中で相手が協力しているときは，実にフレンドリィで笑み満面なのだが，いったんその枠組みをはみ出して，自分に不利益を与えるようになると，とたんに人間が変わる。態度を一変させて，本人の失望感をはるかに上回る幅の落差でもって，怒りを表明し，必要に応じて反撃や制裁にうってでる。

◇

前置きがえらく長くなったのであるが，なんのことはない，またまた湾岸戦争の話である。湾岸戦争を通して，アメリカの強引さと身勝手さを強烈に見せつけられた。そして，憤りを隠せなくなったひとりなのである。

誤解のないようにあらかじめ書いておくが，なにもイラクの行為を認めていたわけではない。クウェートでの大惨事と暴虐は論評をはるかに超えているし，いっていたこととやったことがまったく一致しない。

しかし，リンケージをはじめとして，イラクが唱えた聖戦論の多くがまともに聞こえ，思わず肯定したくなる。それほど，ブッシュ大統領やアメリカ国民のいう「正義」はある側面しか示していない。いやになるほどあからさまで，自己中心的身勝手さに満ち満ちている。

別にあらためて書くほどのことではないが，イラン・イラク戦争における介入，レバノン虐殺におけるイスラエル支持，リビア空爆，パナマ侵攻……，ことごとく自己規定による一方的な視点から行動原理を決めて動いている。しかもそれに同調しない者は，「悪」であるとする単純明快な論理。

こうした側面とあわせて見落としてはならない，アメリカ人の持つもうひとつの行動理念が，一度決めたらめったなことでは変えない，という点である。

昔，外資系メーカーの日本人社長氏と話していて聞いたこと。

「振り上げたこぶしを収めるとか，矛先を収めるという意味の英語は使わないんですよ。引金を引いたあとは，弾は飛ぶだけなんだから」

本当にそういう意味の英語のイディオムが存在しないのかどうかは知らないが，いかにもいいえて妙である。日本人のように，宣戦布告してから，どうやって事態を収拾していくかを考えるような抜けたことをしないのは，子供の喧嘩の仕方を見ていてもわかる。

それだけ単純直情型だといってしまえば，そこまでなのだが，困ったことに，世界は自分たちが動かしている，という強烈な自負心がエスカレートする一方なので，手のつけようがない。

とりあえずは，湾岸戦争を総合論評する意味でも，太平洋戦争での日本の追いつめられ方をきっちりとトレースしておく必要があるように思う。半世紀前の出来事とはいえ，まだうっかりと歴史の1ページにしまい込んではならない問題であろうから。

それに湾岸戦争の終盤戦でのソ連による調停工作介入以後の，アメリカ側の積極的な軍事行動は，妙に当時と酷似する部分があるはず。

◇

もっとも，当座自分でやっておかなくてはならないことといえば，その程度だろう。

「アメリカに憤りを感じるのなら，なにか具体的な反米活動でもするの？」

とでも聞かれたら，悲しいことながら，なにもできやしない。あとはせいぜい「ルパン3世」(劇場アニメ版第1作) の次元大介よろしく，「Macintoshをほしいと思っこともあったが，もうこれっきりだ！」とでも決意 (!?) する程度。それ以外では「沈黙の艦隊」を読んで，某代議士と同じく憂国の情にひたるぐらいが限度か。

それも悲しいところはあるんだけど……。

ちなみに，国産パソコンでもCPUがアメリカ製なら同じじゃないか，って声も聞こえてきそうだけど。残念ながら，ぼくの「国産標準パソコン」は，米国の裁判所が認定した日本オリジナルCPUってやつを使っているのであった。

うーん，そろそろ限界なんだよなぁ。

復刻版面白玉手箱

知っている人は知っているでしょうが，この連載を始める前に本誌で書いていた連載は「絵夢絶登面白玉手箱」(自分ながらよくもまあこんな当て字を）というタイトルがつけられたものでした。いまのように毎回ひとつのテーマについて書くということもありましたが，基本的にはいくつかのそれほど関連のないテーマを合わせてひとつの記事にしていました。今回はそのような感じを復活させ，いま個人的に心が動かされているものをざっくばらんに書き留めてみました。

HARDWARE

ブックパソコン

とにかく小さくて軽いマシンがほしいという気持ちが強まってきました。小さいマシンというと強烈で印象的だったのは，NeXTが発表されたときにコメントを求められたMITのメディアラボ所長が話した表現です，「3～5年後にはパソコンはゴルフボール並みの大きさになっていなければだめだ」(文献1)。ゴルフボールまでも小さくなったマシンといわれても，入力や出力の問題も含めていったい何ができるのだろうかということはまだまだ未知数といわざるをえないでしょう。

とりあえず使えるワープロ機能＋キーボードのついた軽量マシンがほしいのです。そして，そのイメージに近そうなマシンがいとも簡単（そう）に発売されたのです。富士通の出したFMR-CARDです。なんと990ｇという軽量でキーボードもそれなりのものがついているというのです。そして，アルカリ乾電池2本で8時間駆動できるというのもなかなかです。

そうはいうものの，気になるのは年内に出るといわれているMacintoshのブック型モデルです。すでに発売されているポータブルMacとは違い，このモデルには何か驚くべき機能が盛り込まれているのではないかという期待感を持たせてくれるのです。なぜかというと，現在のApple社長のスピンドラーのインタビューの中につぎのような表現があるのです，「ノートパソコンに対するわれわれの姿勢について明確にしたいのは，単に小さくて軽いだけの製品は作らないということです」(文献2)。

小さいだけのマシンがほしいと思っていた気持ちが少しゆらいできてしまいました。皆さんはどうでしょう？　僕はとにかく迷ってしまっているのです。両方とも別の種類のマシンだと割り切って2つとも買うのがいいのでしょうが。

リムーバブルハードディスク

僕のMacにつながっているハードディスクはスーパーマック社のデータフレームXP30というものです。もちろんどのマシンでも同様ですが，特にMacでは，日本語環境を整えフォントに凝ったりしだすとあっという間にディスクがなくなってしまいます(その容量がたとえ100Mバイトあったとしてもです)。さらにそれに加えて恐ろしいのがハードディスクのクラッシュです。実は昨日もある学生の研究成果がパーとなる事故が起きています（僕も無視できない被害を受けた)。

ハードディスクの容量不足とバックアップ問題を一気に解決する製品，それがリムーバブルハードディスクです。これは要するにフロッピーディスクのようにメディアを入れ替えることができ，しかも1枚で(僕の買った製品，マイクロネット社のMR-45では）45Mバイトも容量があるというものです。さらに，アクセススピードが25msと普通のハードディスクの中でも比較的高速な部類に入ります(20msのものもある)。このリムーバブルを購入したばかりですがかなり気に入っています。そして，「これからはリムーバブルだよ」などと来る人に対してしつこく自慢するのでした（自慢ついでに「速いだけでなく安いっ」)。

GAMES

A列車で行こう PartIII

僕自身はちらりと画面を見た程度なのですが，ふだんはテレビゲームなどには熱中しないようなタイプ（たとえば毎日マシントレーニングを重ね，しかも歩く音がいちばん大きいような）の人までもが，このゲームに熱中しているのです。シムシティーまではなんとかついていけた僕は，熱中しまくっているY氏に質問しました。

Q1.何があなたをそのように何日も徹夜させるのですか？

A1.簡単に終わることのできない自己満足の追求があるから。

Q2.ゲーム時の快感を教えてください。

A2.総天然色自動箱庭の喜び。

Q3.次作が出るなら何を期待しますか？

A3.自分の作った都市にもっとのめりこみたい。列車を選択して，それに実際に乗ったときに見える風景をリアルタイム3Dで眺めてみたいですね。実際に暮らしている人々の生活を見たりもしたい。不用意に撤去されてしまった列車に乗っている1,200人の乗客はきっとキョーガクに身を打ち震わせて……(これ以上は危ないので終わり)。

パズル本4冊

最近どういうわけか立て続けにパズルやクイズの本を買ってしまいました。まあそれもたわいのない本といえばいえますけれど，読む場所や時間などによっては格好のひまつぶしになります。それらの本をどこに常備しているかということは，ここではあえて隠しておくとしまして，その4冊は以下のとおりです。

樺旦純：パズル天才塾（三笠書房）
多湖輝：頭の体操第12集（光文社）
芦が原伸之：1年遊べるパズルの本（ごま書房）
デニスシャシャ：ドクターエッコ――奇妙なパズルの依頼人たち（日経サイエンス社）

「1年遊べるパズルの本」はたしかに数はすごく多いのですが，小学生のなぞかけみたいなので水増ししているというような印象が読んでいるうちに強まってきて，1年間楽しむことはできませんでした。

「パズル天才塾」は，ただ機械的に難しい図形の問題を作りましたというのが目立ち，答えを見た場合でも「なるほど！」という印象がしないようなものが多く，いまいち読後感がよくありませんでした。

一方，多湖氏の本は往年のベストセラーとしての貫禄があり良質なものも多く安心して読めます。風呂の中でぼんやりと考えているとわかってくるというぐらいの難易度のものが多く，わからなくて答えを見ても落胆するようなものはほとんどないといってよいでしょう。

異色なのが最後の「ドクターエッコ」です。パズル小説とでもいうのでしょうか，エッコという天才的な数理学者が依頼人たちから与えられた難問を解いていくという形態をとっています。でも，小説の比重よりパズルの比重が大きく，数としても約40

も含まれています。パズル小説という形態自体はそう新しくないような気もしますが，斬新なのは筆者が並列処理や人工知能を研究している教授であり，パズルも計算機科学の色が濃いというところです。訳者のひとりもソフトウェアの研究者です。ソーティング，組み合わせ理論，ネットワーク，論理学，暗号などの計算機科学における基本的な問題を，まるでそのような学問的体系などとは関係ないように気軽にお遊びのように触れられるという点でお勧めです。

SOUNDS

セントギガ

妙な放送が始まっています。静かな音楽に川のせせらぎなどの自然音をミックスしたものが主で，たまに小さな声でささやくように発せられる語りが加わる放送です。これは衛星放送の3チャンネル(WOWOW)の副音声として流されているPCM放送（画面はなし）です。

僕などは「最近テレビはつまらん」などとついこれを選んでしまい，よく聞いています。水の流れる音がとても好きなのですが，たまに寒いときなど一層寒さがつのってしまうのもまたおつなものです。4月から有料放送になってしまうのが残念です。

ベルベットアンダーグラウンド

ベルベットアンダーグラウンドという伝説のバンドをご存じですか？　1960年代末ごろに登場というのですからずいぶんと昔です。ポップアートの奇才アンディ・ウォーホルがからんで誕生したグループです。一部には熱狂的なファンがいたそうですが，あまり売れなかったようです。しかし，その後のパンクやニューウェーブなどに大きな影響を与えたといわれています。

僕自身も興味があって，昔レコードを注文したことがあるのですが，廃盤ということでだめでした。ところが最近，全部持っているというきわめて珍しい男がいまして，その人からCDを借りています。前衛的というのでよっぽどわけのわからないハチャメチャなものかと思っていました。ところが，期待に反してというか，実に普通なのです。しかし，何度も聞き続けているうちに中毒になってきてしまいました。そして，パンクやニューウェーブなどに大きな影響を与えたということがまったく嘘でないことがわかってきました。こういうものには，近寄らないほうがよいでしょう。

MOVIES

バグダッド・カフェ

アドロンという人の作った映画です。とても気に入ってしまい，何度かビデオで見ました。映像も音楽もいいです。バグダッド・カフェという名の宿つき喫茶店に来た掃除魔おばさんのちょっとほのぼのとした話ですが，独特の雰囲気が見るものをひきこみます。バグダッドというといまでは流血の町になってしまいましたが，単にガソリンスタンドつきということを表しているのかもしれません。3月末，この別れの季節にちょっと「おセンチ」になっている人にも勧められるでしょう。

ストレート・トゥ・ヘル

アレックス・コックスという一見いかれたチンピラ男の作っためちゃめちゃな映画です。とにかく皆死んでいきます。「ストレンジャー・ザン・パラダイス」のジム・ジャームッシュ監督やクラッシュのジョー・ストラマー，映画「ブルーベルベット」の変態男（名前は忘れましたが）など超個性派の出演も見逃せません。でも，まあそれほど勧めません。僕のようにお金を払って見にいくのは少し間抜けかもしれません。

BOOKS

科学的ロマンス？

奇妙な本を出すことで一部では有名な国書刊行会が出版した本が，C. H. ヒントンの「科学的ロマンス集」です。科学的ロマンスという言葉の響きに惹かれて衝動買いしました。でも，Romanceという言葉は，英語では日本で使われている恋愛がらみの意味よりは，伝奇小説(現実離れした小説)という意味で使われるほうが多いようです。この本に含まれている3つの小説（もどき）もいわゆるロマンスとは無縁です。

さて，最初の「第4の次元とは何か」と，次の「平面世界」は，それぞれ4次元世界と2次元世界について，「本当にそのような世界に住んでいるとしたら？」ということが書かれています。大真面目な科学教養書にむしろ近いといえるかもしれません。

3番目の話は毛色が違い，王がこの世の中を神のように文字どおりゼロから作っていくという話です。ある生命を作るときに1000の快楽と998の苦痛をその生命に与えると，快楽が少し上回っているので命が吹き込まれるが，逆に王自身が差し引き2の苦痛を味わっていくというへりくつみたいな話です。この話は2部構成になっており，前半がこのような物語で，後半で前半の物語を因果関係，エネルギー保存の基本的な話を交えながら解説しています。

この書物は1800年代後半に書かれたものであり，そのためどのような読者を想定してどのような目的で書かれたか，いまいちピンとこないところがあります。2次元や3次元，あるいはエネルギー保存という概念に対しては100年前，どのようなイメージが存在していたのでしょうか（あるいは存在しなかったのか）？　でも，そこらへんが，国書刊行会発行，あるいはボルヘス編纂ということも手伝ってか一種奇怪な雰囲気が全編にわたって漂っているのです。

知恵の木

最後に紹介するのが「パソコンを思想する」(翔泳社)という本です。この種の本は最近案外多いのですが，これは読みやすくてしかも内容の濃い優れた本です。執筆者10人とインタビュー5人です。特に異色なのは漫画「ぼのぼの」のいがらしみきおでしょう。といっても，彼の文章の内容に関してはノーコメントです。

岩谷宏，津野海太郎，外山寛の3人の文章は特に面白いものといえましょう。津野海太郎の「マニュアルに憧れて」の中で紹介されているウンベルトとバレーラという人たちの書いた「知恵の木」という本が魅力的に，しかも謎めいて書かれていて，欲求不満になります。ぜひ読んでみようと思っています。「この本の全体は，いわば確信への誘惑へと身を委ねてしまうという習慣を，いったん中断してしまおうという呼びかけだ」というのですが，想像つきますか？　しかも本の中の絵の一部に，人の手，ひげ男，はげ男など，計算機の画面にあるアイコンみたいなものがあり，いろんなところに現れてくるというのです。

参考文献

(1)「NeXTの実像を探る」，日経エレクトロニクス，1989.1.23 (No.465), pp. 128-142.
(2)「インタビュー：M.H. Spindler氏」，日経エレクトロニクス1991.2.18 (No.520), pp. 191-193.

第58回

猫とコンピュータ

脳ミソをくれぇ

Takazawa Kyoko
高沢 恭子

○○ショウ，××コンテストなどイベントがあるたびにかりだされるキョウコさん。当然愛機も一緒に，です。でも，そうして回を重ねるごとに梱包は簡略化され……。今回はそれにまつわる悲劇のお話です。

思えばあのころが臨終だった。いまは死んでいる状態で，なにかの拍子にフラフラと動き出しては，また止まる。まるでゾンビかバタリアンのように。

あっというまに歴史上の「マイコン」になってしまったTK-80だが，わが家のテクノ書斎では貴重なルームメイトの1台であり，ふだんは静かに眠っているけれど，動いてほしいときにはいつでも目をさましてくれた。

彼がしごとをさせられるのは，おもにパソコンのイベントのときで，実用としてというより，ショウのアイドルとしての出演をもとめられていた。「元祖パソコンでーす。まだ元気で走れますよぉ」といったふうだ。

あるときは「ペントミノ」のパズルを延々と解きつづけたり，あるときは「人工無脳」になって問答の相手をしたり，またあるときは来客を知らせるセンサーになって，イルミネーションを点滅させた。

骨組みも内臓もむき出しのTK-80だから，イベントが近づくと，しごとに合わせて体の中を遠慮なくいじられた。あっちをはずし，こっちをつなぎ，それでもけっこうケロリとしてよく働いたものだ。

15年選手の彼をもっと気づかってやればよかったと，いまにして思う。

アキバから池袋へ

昨年の10月，雑誌でホビーマイコンショウの開催を知ったという，V企画のナカジマさんからお電話があった。

「ぜひお願いしたいことがありまして，会場におうかがいしたいのですが」。ショウの当日，ナカジマさんはやってきた。

「池袋のサンシャインビルで毎年開かれるパソコンフェアなんですけど，ことしはパソコンの活用コンテストを予定しているんですが，出品作があつまらずに困っています」

そこで，マニアの作品があつまるホビーショウに期待をかけたらしい。

「グループの方たちの作品を4点くらい出品していただけたらと……」

ナカジマさんは会場を一巡して，候補の作品をいくつか挙げた。「TK-80もお願いします。これは目玉になりますし」。

ところで，コンテストの当日は作者も出場して，作品についての説明をしなければならないのだそうだ。みんな超多忙な人たちばかりで，このホビーショウですでに相当な無理をしているのに，サンシャインビルでのパソコンフェアはわずか6日後だという。

それでも困っているナカジマさんを見捨てるわけにはいかない。なんとかスケジュールをやりくりして，4つの作品とそれぞれの作者を，10月20日の池袋に送り込むことになった。

「パソコン活用コンテスト」に出品するのは，シンヤさんの「簡易CAD」，クガさんの「バック・グラウンド・ソフト」，オザキさんの「宇宙線測定」。それに夫のTK-80を使った「BSマージャン」。まずは事前にマシンを会場に運ばなければならない。これはやっぱり宅配便だ。ホビーショウを終えた翌日から発送の準備となった。

このころもうTK-80は死線をさまよっていたのかもしれない。でもそれに気づくには日程があわただしすぎた。それよりも，無理な頼みをしたほかの3人のマシンが気になっていた。ナカジマさんが管理してくれるとはいえ，人出もケタちがいに多いパソコンフェアの会場に，ほかの出し物とまぎれたりせずに到着させなくてはならない。

折しも，クガさんの使うPC-8001用のカラーCRTがどうしても調達できないという連絡が入り，わが家のものを送ることになった。

TK-80と白黒モニタ，キーボード，それにカラーCRTは，まったく無造作な梱包で発送された。いままでに，マシン類を宅配便で搬送しても，故障やトラブルは一度もなかったのだから気軽なものだ。ただ，発送の前に作品のデモをビデオ撮影しておいたのが，虫の知らせというものだったのだろう。

クサヤのくん製も

「活用コンテスト」はパソコンフェア開催中の3日間とも行われたそうだが，参加をたのまれていたのは最終日だった。

男女ペアの司会者の進行で，ひとりずつ作品とともに登場。製作までのいきさつを説明したり，司会者の質問に答えたりしながら，実演してみせる。審査をするのは来場者で，個々の観客席にあるパソコンから番号キーにより投票。得点の集計が，ステージ上の電光掲示板に表示され，優勝者にはカップが贈られる。

夫の作品はTK-80を使った自作のソフト，「日本初の4人マージャン」(昭和54年にI/O誌に発表したもので，つぎつぎ新しい機種に移植，のちに「精彩グラフィック麻雀」としてヒットした)。

説明を終えていざプログラムを走らせようとしたら，どうもようすがおかしい。家でCRTを撮影しておいたビデオが，ここで役に立った。

あたかも予定していたかのように，「時間の節約のためにビデオでお目にかけます」と，うまくその場をのがれた。

ほかの出場者は，パソコン通信のネット

の情報から,「クサヤのくん製」をじっさいに製造したという,大手自動車会社につとめる若い男性。パソコンの生きた活用としてユーモラスだった。同じくパソコン通信で,「CINEMA-NET」という日本で初の映画専門のBBSを紹介した男性。会場からアクセスをこころみて,書き込まれた記事の一部を披露した。ネットの内容は本として出版もされたそうだ。

全8点の作品の中で,高得点でカップを手にしたのは,血圧管理のプログラムを作製しておおいに役立てている62歳のご婦人と,市販の音楽ソフトを作曲に活用しているプロの音楽家。演奏のデモは,なんといってもわかりやすいし華やかだ。

夫は「日本最古の4人マージャン」と,骨董品の機種が珍重されたのか得票が多かったが,仲間の3人は,「簡易CAD」や「宇宙線測定」など高度な知識を必要とするにもかかわらず,一般の人たちの理解の範囲でなかったのか,得票は低かった。とはいえ,もとよりシロウト採点の遊びなのだし,ナカジマさんに協力できればそれでいいのだった。

簡易包装のススメ

コンテストが終わるやいなや,いそがしい面々はふたたびマシンの梱包を始めた。こんどはそれぞれの自宅にあてて送り返さなければならない。3人の方たちはともかく,夫の荷造りは最低限の簡略さだったらしい。発送の翌日には着くはずの荷物が2,3日たってもわが家に届けられない。さすがに気がかりで,ナカジマさんに問い合わせてみたところ,すこし口ごもりながら,

「じつは梱包が悪いので責任が持てないと,宅配の会社から一度もどされまして,荷をつくりなおしたんです。まもなく届くと思いますが……」

ナカジマさんにおわびとお礼をいいながらも,じっさいには私たちはなんの反省もなく,「そんな心配はムダなのにねぇ」とまだタカをくくっていた。

荷をほどく手間と時間は,荷造りの時間の自乗くらいにも比例する。つまりていねいなほど損失が大きくバカをみる。にもかかわらず,荷造りの質と損害のあいだにはほとんど関連がみあたらない。8回ほどの引っ越しでただ一度,タタミ1枚ほどの大きさの鏡のコーナーが,1辺1センチくらいの三角形に破損したのは,私たちではなく,プロの業者の梱包によるものだった。

それでもパソコンや機械類についてだけは,初めのうちは神経を使っていたものだ。でも,だんだんそれも楽観的な主義に変わってきた。かならずほどくことがわかっているのだから,梱包はかんたんがよい。こうしたイベントなどでマシンの運搬をくりかえすうち,精密なはずの機器に対して大胆になってしまったともいえる。

そしてTK-80は帰ってきた。

コンテストの会場では動かなかったけれど,わが家なら安定してまた元のとおりになるだろうと,夫は調べ始めた。

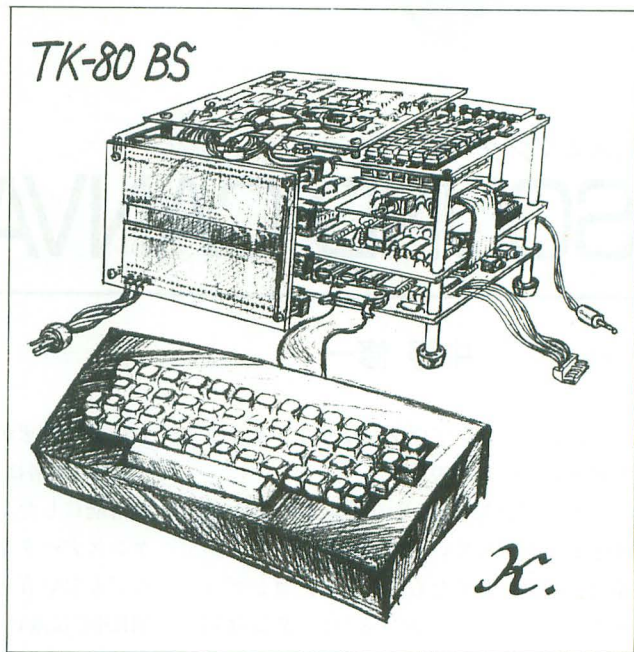
ところがなんとも変な調子だ。まずBASICが出てこない。ちょっと動いたと思うとすぐ止まってしまう。だめかと思っていると,またすこし動く。こんなことは初めてだ。でも,きっとなんとかなるだろうと,ここでもまだ楽観的な私たちだ。

TK-80はそんな軟弱なマシンじゃない。構造はシンプルなんだ。運搬や荷造りについての心がまえをあらためるなんてこととはちがう。

脳ミソのリスト

BASICが出てこないところをみると,原因はたぶんCPUのあるTK-80のボードだろうということになった。モニタROMが悪いとしたら,こんな古い部品は古ぅーい仲間しか持っていない。

古い仲間はたくさんいて,TK-80を持っている人も多いけれど,スペアの部品なんてまず望めない。部屋の隅か,押し入れにたいせつにしまってあるTK-80は,分解すれば,自分も死んでしまう。

「そうだ,イシイさんのところには,コバヤシ先生のが置いてあるはずだから,ひとつ借りればいい」

さっそくイシイさんに連絡してボードを送ってもらい,わが家のボードと交換してみた。ところが変化がみられない。それではBASICのROMがついたBSボードに原因があるのではないか。

BASICのROMにはLevel I用と,Level II用がある。どちらが悪いかつなぎかえてたしかめてみようということになった。切り替えボードも使われているので,原因の追求にはいく通りものテストが必要だったが,どうやらLevel IIのROMに異状があるらしいとつきとめた。

「イシイさん,Level IIのROM,ちょっと貸してよ,TKのボード返すから……」

博物館ものの貴重なマシンの部品だから,いくら古い友人でもたやすく手渡すわけにはいかない。それにROMをとりだすにはバラさなければならない。デザイナーのイシイさんは,しごともいそがしいのだ。

「ホシノさんに聞いてみてよ,ダメならなんとかするから」

そりゃそうだ。脳ミソなんか人にあげるものじゃない。

「ねぇねぇ,パパ,オオモリさんも,アダチさんも持ってるでしょ。それから日暮里のオミさん,倉敷のナカタさん……」

「川鉄のアキタさんも,もしかしたら木場のコヤマさんも持ってるナ」

誰の脳ミソをもらうか,黒いリストを2人でつくり始めた。これからはマシンの運搬にはもうすこし慎重になるだろう。

新製品紹介

GSスタンダード仕様

SOUND CANVAS SC-55

Nakano Shuichi 中野 修一

ついにパソコンで標準となりそうなサウンドモジュールが現れた。

これまではMT-32系の楽器が事実上標準として扱われていたのは周知のとおり。単なる楽器としてならともかく，コンピュータミュージックにおけるMIDI楽器は新たなメディアとしての可能性を秘めている。これはMIDI特集のたびにいっていることだ。MIDIデータフォーマットは大理石のように美しい統一ぶりを見せている。しかし，データ自体にはまだ音色の標準化という問題を抱えていた。こちらはこれまで藁ほどの統一さえ見られないものだった。

それがいま統一へ向けて大きく一歩を踏み出した。それがRolandの提唱する「GSスタンダード」による音色セットの登場，そしてGSスタンダードを採用したSOUND CANVAS SC-55の登場だ。

GSスタンダード

話は変わるが，本誌で発表したOPMDではMIDI楽器へのデータ出力の際に音色番号の変換，ボリューム/オクターブの調整，ドラムセットの統一までやってのけた。重いOPMDRV上でさらにPCMまで鳴らして，そのうえこのような処理まで抱え込んでいる。なぜか？

MIDI規格はさまざまな機器を相互に接続可能にした。しかし，ある楽器用のシーケンスデータをほかの楽器に適用するととんでもない音色で演奏されてしまうのだ。MIDIでは避けられない問題だった。

MIDIという規格は演奏データの互換性をかなり高いレベルで実現している。シーケンスデータの互換性を阻んでいるのはごく些細な規格洩れ（？）にすぎない。これはあまりにもったいないことではある。

OPMDでは各音源ごとの音色を自由に再配置できるように3種類のテーブルを用意し，メーカーでまったく実現方法の異なるリズムキットも力技で変換している。

どうして音色の問題が見逃されてきたのだろうか？　答えは，そういう要求がなかった，という事実に帰着する。MIDIを使った音楽活動といっても，ひとりの音楽家のまわりで統一されていればよかった。それぞれが複数の音源を使いさらに生楽器とのミキシングなどを重ね，作品というのは最終的な「音」としてしか存在しなかったからだ。

コンピュータはそれを変えていきつつある。まず，何種類もの音源を駆使するものではなく単一の音源で「デスクトップミュージック」というジャンルが出てきた。これまでも同じマシンで同じ音源，同じソフトウェアを使う限り演奏データは共有できた。音源の種類がわかっていれば変換することも不可能ではないだろう。途中のシーケンスデータ自体に価値が生まれ，それ自体で流通し始めたのだ。

これによってMIDIはこれまでより遥かに大きな可能性を示すようになったといえる。MIDIのあり方さえも変えてしまうほどに。

ここでMIDIでの音色標準化の発想が見えてくる。そしてゼネラルMIDIによって音色統一という新たなMIDIの世界が登場したのだ。その音色標準化をさらに拡張したものがGSスタンダードだといえる。

GSスタンダードはRolandが提唱する音色セットおよびMIDI音源の扱い方の標準化を目指した規格だ。残念ながら現状ではあまり実体はわからない。ゼネラルMIDIにアッパーコンパチで16ティンバーのマルチ音源で標準の128音とドラムセットを持つという以外のことは情報公開されていない（なぜだ？）。

さて，このような動きがなかなか現れなかった最大の要因としてMIDI規格自体の弱さが挙げられるだろう。MIDIでサポートされているプログラムチェンジは128種類，すなわち音源は128音しか持つことができなかったのだ。ところが，コントロールチェンジの0番が音色バンクの切り替えとして設定されたことにより，MIDIは一気に128バンクの音色セットを持つことができるようになったわけだ。GSスタンダードといっても標準化するのはそのうちの1バンクだけで，楽器全体の仕様を制限するようなものではない。標準化のための付加機能と考えればいいだろう。

音源モジュールSC-55　69,000円（税別）
ある程度は音色のエディットも可能。リモコンも付属。

どこも同じ音色では個性がなくなるのではないか？　という疑問も出てくる。

世界中にオーケストラはたくさんあるが人数も違えば，演奏者の技量もマチマチだ。出てくる音も異なる。それでも，同じ譜面を渡せば同じ曲が奏でられる。決して同じ演奏ではないかもしれないが，「楽譜」というシーケンスデータはオーケストラという「音源」によって新たな価値を与えられることになる。音源が違えば同じ音色名でも実際の音色は異なる。同じ土俵に立ってこそ個性が明確になる，ともいえる。

GSスタンダードが規定しているのは音色だけではないという。演奏に必要な情報はこれまでもコントロールチェンジとして規定されていたが，その解釈は機器によってマチマチだった。たとえば，パン（左右のステレオ定位）ではあるものは右・左・中央の3段階であり，ほかの機種では左右128段階だった。また，ベンド幅にしても最大1音だったり，1オクターブ固定だったり，可変だったりとさまざまだ。

そのほか，これまではどうしてもエクスクルーシブメッセージに頼らなければならなかった部分というのを極力減らすように留意されているという。このような配慮があって初めてシーケンスデータは機種を超えて広がっていくことができるのだ。

SAUND CANVAS SC-55

では，SC-55の機能解説といこう。

見た目は写真のとおり，ハーフラックサイズのコンパクトな音源モジュールだ。

16マルチティンバー，最大発音数は24パーシャル，エフェクタとしてリバーブとコーラスを内蔵。

パーシャルというのはRoland系の音源のウェイブジェネレータまわりの部分で，これをいくつか組み合わせてひとつの音色を構成する。よって24パーシャルというのは最大24音ということになる（すべて1パーシャルの音の場合）。ここで，24パーシャルが多いか少ないかということが問題となる。

たとえばFM音源の8音はおいておくとしても，KORGのM1でも最大16音だ。多くのシンセサイザでは16音程度の同時発音が一般的といえる。MT-32と比べても，ひとつの音色で使用するパーシャルがMT-32に比べて少ない。ほとんどが1，2パーシャルで構成されている。たくさんのパーシャルを使ったほうが複雑で豪華な音が出せる。これは当然。音質は大丈夫だろうか？聞くところでは，MT-32系より良質な音を実現しているという。「パーシャル」というのはFM音源でいうとオペレータにも相当するが，FM音源では源波形がもっとも単純なサイン波，LA音源はもっと複雑……。要は源波形しだいということだ。ちなみにM1をRoland風に見れば最大でも2パーシャルだ。

さらに増設（並列？）により48パーシャルも可能とくれば，これは必要十分な仕様だといえるだろう。

音源まわり以外に目を向けてみよう。

MIDI INを2つ持っている。音源モジュールという性格を考えると，ひとつはコンピュータやシーケンサに，もうひとつは鍵盤つきの楽器に接続するという選択がベストだろう。これらは内部でミキシングされるので両方同時に使うこともできる。

同様にAUDIO　OUTとともにAUDIO INの端子を持っている。音源になんで？という疑問はもっともだろう。音声信号によって音を制御して……というWAVESTATIONのようなことができるわけではない。単なるミキサーなのだが，これがパソコンにとっては非常にうれしい機能だといえる。FM音源やAD PCMなどの本体からの信号を入力すればMIDIの出力とまとめてアンプにつなげる。これでミキサーを買わなくても演奏と効果音がバラバラになったりすることもなくなるのだ。

スタンダードMIDIファイル

同時に発表されたSOUND BRASHはスタンダードMIDIファイルに対応した3.5インチFDDを内蔵している。スタンダードMIDIファイルはFD上で実現されたMIDI規格ともいうべきもので，多くのシーケンサやATARI ST，Macintoshの有名ソフトなどが対応している。大雑把にいえば，データの入ったフロッピーさえあれば，どのマシンでもそのまま読み書きできるわけだ（演奏の際はFORMAT 0/1などの違いもある）。

これはまたシーケンスデータの共通化を加速する材料ともなりうる。たとえばサン・ミュージカルサービスが行っているMusicstudioのデータ曲集がある。みゅーじくんデータ集やMUSIC PRO-68K101曲データ曲集はその名のとおり「データ集」だが，こちらは一流ミュージシャンによる「アルバム」ともいっていいだろう。

音色共通化によりこれがさらに一般化すれば，まさにCDなどのように，シーケンスデータを入れたディスクがそのままレコードのようにやり取りされる……。

8cmCDに入る音楽はせいぜい20分くらいだが，8cmCD-ROMには辞書数冊分のデータが入るのだ。しかも生演奏。考えようによっては新しい音楽出版形態の誕生である。

とにかく，X68000/Musicstudio/MT-32などという環境に依存せず，こういったものがグローバルに展開する可能性が出てきたわけだ。

問題はこのGSスタンダードの規格がまだRolandだけのローカルなものにすぎないことだ。KAWAIなど，YAMAHA系を除く国内メーカーは採用の模様？　との噂も聞くがまだ詳しいことは未確認だ。海外でゼネラルMIDIの認定は受けたといわれるが国内のMIDI協議会にはまだ認定されていない。

とりあえずいまは，いち早くこのような標準化に乗り出したRolandの英断に拍手を送りたい。

MIDIシーケンサSB-55　59,000円(税別)
ディスク上のデータをCD感覚で演奏できる。
曲途中からでも常に正しい演奏が可能。

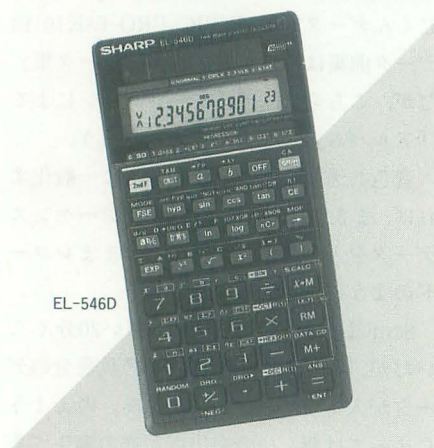
PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

ハイスピード関数電卓

EL-546D/540D/506D

シャープ

EL-546D

シャープは関数電卓の新シリーズ（ハイスピードシリーズ）として，同社従来機の約3倍の演算スピードを持つ「EL-546D/540D」と，わかりやすいドットマトリクス式ガイダンス表示を採用した関数電卓「EL-506D」の3機種を発売した。

「EL-546D」はわかりやすいガイダンス表示，複素数計算機能（複素関数，カッコ機能つき），1次，2次などの6種類の回帰計算機能を搭載した中級機，「EL-540D」は2進，8進，16進論理演算機能を搭載した普及タイプで，どちらも分数計算，時間計算などの基本的な関数も充実している(それぞれ134/80関数機能)。

「EL-506D」は入力，演算結果がわかりやすいドットマトリクス式ガイダンス表示，1次，2次回帰などの6種類の回帰計算，数値積分，時間計算機能など充実した機能を持つ中級機である。

また，「EL-546D/540D」は本体組み込みの太陽電池，およびバックアップ用のリチウム電池を使用，「EL-506D」はアルカリマンガン電池を使用する。

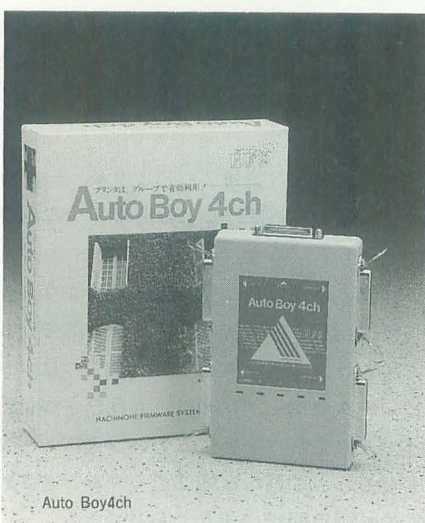
価格は「EL-546D」が5,800円，「EL-540D」が4,000円，「EL-506D」が4,800円（すべて税別）となっている。

〈問い合わせ先〉

㈱シャープ　☎03(3260)1161，06(621)1221

自動プリンタ切り替え器

Auto Boy4ch

八戸ファームウェアシステム

Auto Boy4ch

八戸ファームウェアシステムは自動プリンタ切り替え器「Auto Boy4ch」を発売した。「Auto Boy4ch」はパソコン4台対プリンタ1台で使える自動プリンタ切り替え器である。プリントアウトが開始されると，どのパソコンのデータを印字するのかを「Auto Boy4ch」が自動判別し，瞬時に切り替えを行う。印字終了後，次のパソコンのデータを自動的に印字開始するので，手動切り替えタイプのような煩わしい手間を省くことができる。

また，LEDランプ表示により，4台のパソコンのうちのどのパソコンが印字中なのかがひと目で判別できる。110（W）×175（D）×28（H）mmというコンパクトサイズなので置き場所にも困らない。

価格は39,800円（税別）。

〈問い合わせ先〉

八戸ファームウェアシステム㈱

☎011(716)3815

電子手帳用ICカード

「松本亨の凱旋門」

水谷電機工業

PA-9500とPA-Z07

水谷電機工業は株式アナリストの松本亨氏監修のシャープ電子手帳用カード「松本亨の凱旋門<PA-Z07>」を同社販売ルート，およびシャープ販売ルートを通じて発売した。

この「松本亨の凱旋門」は一般投資家を対象に開発されたICカードで，画面表示に従って業績データを入力することにより有望銘柄の発掘計算ができ，また，損益代金計算，CB（転換社債）やワラント債（国内，ドル建）の理論値計算などもできる。

また，監修者の松本亨氏の“注目銘柄”（「今週の注目株」，「今月の注目株」，「長期の注目株」）を専門電話，新聞紙上などで毎週暗号にて発表する。この暗号を本カードに入力することにより，“注目銘柄”の“コード番号，銘柄名”を知ることができる特典機能もついている（このサービスは平成3年3月より平成5年8月までの2年6カ月間，実施される）。

取引手数料や税金などの改定があったときなどに，それに対応してプログラムを変更できるのも特長のひとつ。

価格は19,800円(税別)。なお，2行表示の電子システム手帳「PA-7000」では使用できない。

〈問い合わせ先〉

水谷電機工業㈱　☎03(3253)4341

耐環境性を重視

“マークQ・クリエイト”シリーズ

住友スリーエム

“マークQ・クリエイト”シリーズ

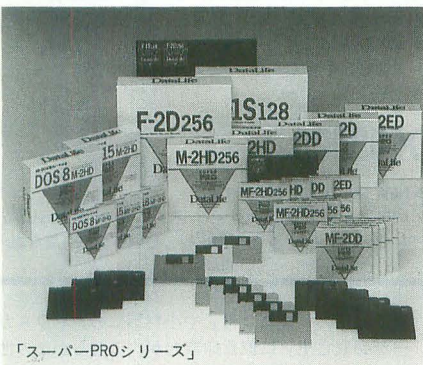
<3M>フロッピーディスク“マークQ・クリエイト”シリーズは，製造工程においてリアルタイム・ディスパージョンフィード・プロセス（塗布液即時供給方式）とエアーフローテーション・プロセス（非接触型磁性層形成工程）を導入し，磁性体密度，分散特性，均一性を向上している。またこれに加え，新バインダーシステムにより磁性層の面強度を従来品比20%向上させている。これらのことにより，本製品は連続10万回におよぶヘッド衝撃に耐え，連続4000万パス/トラックという高耐久性テストをクリアするという耐環境性に優れたフロッピーディスクとなっている。

〈問い合わせ先〉

住友スリーエム(株)　☎03(3709)8111

耐環境性能をグレードアップ

スーパーPROシリーズ

化成バーベイタム

「スーパーPROシリーズ」

化成バーベイタムのDataLife「スーパーPROシリーズ」は，強靱な磁性層を形成する“新CCバインダー・システム”をはじめ，あらゆる環境下での塵埃をシャットアウトする新開発の“高密閉型HSシャッター（3.5インチのみ）”，安定したクリーニング性能の“新UWライナー”など，従来の「PROシリーズ」で培われた技術をさらに高めた高性能フロッピーディスクである。

〈問い合わせ先〉

化成バーベイタム(株)　☎03(3283)6423

16機種分を1パックに

Final SuperPack

エー・エス・ピー

エー・エス・ピーは，MS-DOS対応日本語スクリーンエディタ「Final」シリーズのSuperPackを出荷する。このSuperPackは現在発売中の「Final」シリーズ16機種分を1パックに収めたものである。現在，パソコンの利用形態が1人1台の時代から，1人で複数機種を利用する時代に変わろうとしている。また，エディタの使用用途がプログラム開発に加えて，データファイルの作成，日本語FEPと組み合わせての文章作成と多用化してきている。このようなニーズに対応し，なおかつライセンス形態の簡素化を図るために，いままで個別に販売していた「Final」シリーズを1パック化するということである。初年度販売本数は，20,000本を予定している。

対応機種：PC-9801シリーズ，PC-98XA/XL，PC-98LT，PC-286/386，FM Rシリーズ，FM TOWNS，FM16β，Panacom M，B16/B32，H2020，J3100/J3300，AX，PS/55，IBM-PC，DOS/V(VGA)，X68000

価格は28,000円（税別）。

〈問い合わせ先〉

(株)エー・エス・ピー　☎03(3767)1451

INFORMATION

CD-I内部オペレーティングシステム

CD-RTOS ver.1.1

マイクロウェア・システムズ

マイクロウェア・システムズではCD-Iの内部オペレーティングシステムCD-RTOSのバージョンアップ版(ver.1.1)をリリース開始した。この「CD-RTOS ver.1.1」では，将来CD-RTOSが拡張されたときでも対応しやすくなったなど，新しい機能が加わった。

また，新規オプショナルパッケージとして，従来のオーソリングシステム製品にのみ組み込まれていたCDドライブ，ビデオ，オーディオなど，I/Oまわりのライブラリをまとめた「CD-I・Cライブラリ」，およびライブラリに開発言語であるマイクロウェアCコンパイラをパッケージした「CD-Iプロパック」もあわせてリリースされる。

これにより，オーソリングシステムを持たないソフトハウス，システムハウスでもOS-9/68000システムやUNIXシステム上で，CD-Iアプリケーションプログラムの開発が容易にできるようになる。

〈問い合わせ先〉

マイクロウェア・システムズ(株)

☎03(3257)9000

ビジネスのコミュニケーションを強化

アスキーネット法人会員制度

アスキー

アスキーはパソコン通信ネットワークサービス「アスキーネット」に，企業あるいは企業の部署単位で，団体で入会することができる「法人会員制度」を新設し，会員の募集を開始した。これにより，パソコンやワープロをネットワーク化し，パソコン通信の各種機能をビジネスに活用できる。

この「法人会員制度」は「アスキーネット」の4つのサービス（ACS/DPI/PCS/MSX）に企業や各部署単位で入会するもので，従来1人あたり3,000円必要であった初回登録料が，人数に関わらず法人1件あたり5,000円で入会できることや，月々の使用料金が固定制になるなど，「アスキーネット」のサービスをより廉価に利用できる。

また，利用料金は企業あてに一括請求となるので，予算管理，事務処理の簡略化にもつながると思われる。オプションとして，専用の電子掲示板であるCUGノートサービスを利用すれば，限定されたメンバーによる連絡や会議などを行うことができる。

〈問い合わせ先〉

(株)アスキー　☎03(3486)9661

スーパーファミコン

第3回初心会ソフト展示会に50名をご招待

スーパーファミコンなどのソフトの展示を行う「第3回初心会ソフト展示会」。今回は4月24日から5月6日までの13日間，幕張メッセで行われ，入場料1,000円で誰にでも参加できる。この入場券を50名の方にプレゼント希望欄にNと記入のこと。

FILES Oh!X

このインデックスは，タイトル，注記――筆者名，誌名，月号，ページで構成されています。なんかいいことないかな，なんていってる人，桜も咲いてることだし外に出てみれば？　面白いもの見つかるかもよ。

参考文献
I/O　工学社
ASCII　アスキー
コンプティーク　角川書店
テクノポリス　徳間書店
ポケコンジャーナル　工学社
POPCOM　小学館
マイコン　電波新聞社
マイコンBASIC Magazine　電波新聞社
LOGIN　アスキー

一般

▶実践！　パソコン通信編　POPCOM-NET攻略法
パソコン通信をこれから始めようという人のために，POPCOM-NETへのアクセスを例に，モデムや通信ソフト，パソコン通信電話帳などを紹介，解説している。――編集部，POPCOM，3月号，98-101pp.

▶ハイテク地獄耳
世界初の小型カラーFAX，シャープのJX-5000を紹介。――編集部，POPCOM，3月号，121P.

▶特集MS-DOS手習いガイド
DOSの代表選手「MS-DOS」について，初心者向け解説をしている。初めてDOSを扱うユーザーのためのコマンド解説など。もちろんX68000のHuman68kユーザーにも参考になる。――編集部，LOGIN，3号，227-239pp.

▶NETWORK CONNECTION
中日新聞社が運営する「中日ネット」やアスキーネットPCSの9600bps正式サポートを紹介。――編集部，LOGIN，3号，272-273pp.

▶NETWORK CONNECTION
JR西日本のサービスするネット「JR西日本NET」，PC-VANが始めた「英訳・和訳サービス」，NIFTY-Serveの画像メールなど。X68000のPDSゲームの「悟り」を紹介。――編集部，LOGIN，4号，232-233pp.

▶電話・通話料金のしくみ
NTT電話料金のしくみや，新電電のしくみなどについて図入りでわかりやすく説明する。電話代で泣くことの多いネットワーカー必読。――マイコンネットワーク研究会，マイコン，3月号，117-125pp.

▶Mycom News
48ドットカラー熱転写漢字プリンタCZ-8PC5と，X68000SUPER用の80Mバイト増設用内蔵ハードディスクドライブCZ-68Hの紹介。――編集部，マイコン，3月号，173P.

▶パソコンソフトウェアQ＆A
ソフトウェアの著作権問題をQ＆A形式で探る。個人編や企業編などに分け，比較的身近なコピーに関する問題について考える。――コンピュータソフトウェア著作権協会，マイコン，3月号，258-263pp.

▶なんでも相談室
システムバスと拡張バスの違いについて，コンピュータのバス構造の概念など，図入りで比較的わかりやすく解説する。――加藤，ASCII，3月号，376-378pp.

▶SCSIの使い方
SCSIについて規格から領域確保の話，ファイルのインストールの話，ハードディスク新製品試用記まで幅広く扱った特集。PC-9801やFM TOWNSとの接続を考えている。――編集部，I/O，3月号，79-107pp.

▶こまんど君のCONFIG.SYSって何なのさ？
CONFIG.SYSについてわかりやすく解説してある。システム構築やデバイスドライバの話など。MS-DOS向けであるがX68000にも通じるものがある。――HASEGAWA.H，I/O，3月号，177-184pp.

MZシリーズ

MZ-1500(MZ-5Z001BASIC)

▶廻りゃんせ
画面中央から落ちてくるボールを一番下の円にいれて，時計まわり，反時計まわりに移動させ，画面上部にある3本の柱の中にいれて消す。アイデア秀逸アクションパズルゲーム。――まてりある，マイコンBASIC Magazine，3月号，122-123pp.

MZ-2500(M25-BASIC)

▶KNIGHT
桂馬とびしかできない自分の駒を動かしてコインをとる。全30面。――工藤俊介，マイコンBASIC Magazine，3月号，124-126pp.

X1/turbo/Z

X1シリーズ

▶アルケー
山にこもり障害物競争！　針の山を飛び越えるゲーム。ショートプログラム。――遠藤克之，マイコンBASIC Magazine，3月号，155p.

▶ステッポイ
一筆書きで画面上の石を取るパズルゲーム。――白井建夫，マイコンBASIC Magazine，3月号，156-157pp.

X1＋FM音源ボード(要NEW FM音源ドライバ)

▶keyboard
FM音源の演奏状態を画面上へキーボードとして表示する。音楽演奏ビジュアルプログラム。――登坂宏之，マイコンBASIC Magazine，3月号，173-174pp.

▶グラディウスIII　～Try to Star～
コナミのゲームミュージックプログラム。――たつくん，マイコンBASIC Magazine，3月号，185-186pp.

X1turboシリーズ

▶How To Win
三國志IIを大攻略！　――編集部，コンプティーク，3月号，128-131pp.

X68000

▶NEW SOFTWARE REPORT
MIDIマルチトラックシーケンスソフト「Musicstudio PRO-68K Ver.2.0」を紹介，解説している。――編集部，

新刊書案内

いとうせいこう最初の小説である。というのは，ノーライフキングのあとがきに「次は小説になる」と書いているからで，前作はなんだったのかというと，「物語」である。本書にコンピュータは出てこないが，前作でもコンピュータはただ仮想空間構築文化の道具として存在したのであって，今回も仮想空間構築化の象徴にあふれている。ノーライフキングで夜の街へ飛び出た子供たちがディスプレイ内の仮想空間の仮想へ移行したまま，現実空間が仮装行列している。本書の舞台であるフェイク・ムスリムはイスラム教のイメージを付加した“フェイク”な空間であり，そういった意味で島田雅彦の「ロココ町」や小林恭二の「ゼウスガーデン衰亡史」に似た印象だが，これらがその世界の視点と外の視点の両方が存在するのに対してフェイク・ムスリムの視点しか存在しないピュアさがよりフェイクと快楽に囲まれた仮想空間構築文化を際だたせる。その文化の底ではハウスミュージックやラップが鳴り響いており、グランドビートが鳴らない人には本書のトリップ感がわからないかもしれない。　(K)

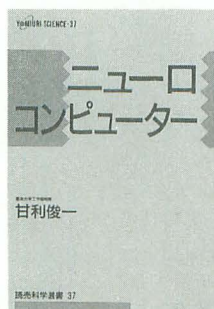
ワールド・エンド・ガーデン　いとうせいこう著
新潮社　☎03(3266)5111　四六版　363ページ
1,500円

書籍の価格は消費税込みです

マイコンBASIC Magazine, 3月号, 74-75pp.
▶NAGDRV情報局
X68000用MIDI＆FM音源ドライバ「NAGDRV」についての読者からの質問や, 情報を紹介。──永田英哉, マイコンBASIC Magazine, 3月号, 88-89pp.
▶誌上公開質問状
"■"記号の打ち込み方, SYSTEMファイル転送のやり方, X-BASICプログラムのリストをプリンタ出力する方法などの質問に答えている。──多田太郎, マイコンBASIC Magazine, 3月号, 91-92pp.
▶Worm
画面上の紫のピラミッドを体当たりして破壊する。モトスみたいなアクションパズルゲーム。──るい, マイコンBASIC Magazine, 3月号, 158-159pp.
▶BARGARTRIS
落ちてくるハンバーガーの材料を順序よく積み重ねてハンバーガーを完成させる。──高橋潤, マイコンBASIC Magazine, 3月号, 160-161pp.
▶Paper War
2人用対戦型ウォーシミュレーションゲーム。──Bee, マイコンBASIC Magazine, 3月号, 162-164pp.
▶空牙 〜VAPOR TRAIL〜
データイーストのゲームミュージックプログラム。要NAGDRV。──KENICHI.O, マイコンBASIC Magazine, 3月号, 189-190pp.
▶TRIGON 〜Strudy Wings〜
コナミのゲームミュージックプログラム。要NAGDRV＋MT-32系MIDI音源。──M.H, マイコンBASIC Magazine, 3月号, 191-192pp.
▶GRADIUSIII 〜Try to Star〜
コナミのゲームミュージックプログラム。要NAGDRV＋MT-32系MIDI音源。──中里和紀, マイコンBASIC Magazine, 3月号, 193-194pp.
▶集中攻略「続ダンジョン・マスター カオスの逆襲」
冒頭部分とKUの道を攻略。──石井弘一＆解せないクン, マイコンBASIC Magazine, 3月号, 229-233pp.
▶Monthly software Hot Press
「マーブルマッドネス」「ボンバーマン」「ファランクス」を紹介。──編集部, POPCOM, 3月号, 20-21pp.
▶ゲームの達人DX
「イメージファイト」と「エメラルドドラゴン」を攻略。──編集部, POPCOM, 3月号, 80-85pp.
▶ミュージック・パビリオン
TVアニメ『らんま1/2』主題歌「絶対！ Part 2」(早坂好恵)のOPMデータ。──編集部, POPCOM, 3月号, 147-150pp.
▶How To Win
「続ダンジョン・マスター カオスの逆襲」,「三國志II」を大攻略──編集部, コンプティーク, 3月号, 112-119, 128-131pp.
▶ロッパー最新情報
「生中継68」,「ファランクス」,「ノスタルジア」,「ラプラスの魔」,「中華大仙」を紹介。──編集部, コンプティーク, 3月号, 228-231pp.
▶GAMING WORLD
新着ゲーム「アトミック・ロボキッド」と, 発売予定の「ボンバーマン」「ノスタルジア」, 移植進行中の「アルガーナ」「ドラッケン」を紹介。──編集部, テクノポリス, 3月号, 24-34pp.
▶攻略ファイト！
4月発売予定の「A列車で行こうIII」と,「続ダンジョン・マスター カオスの逆襲」を攻略。──編集部, テクノポリス, 3月号, 56-69pp.
▶NEW SOFT
「中華大仙」,「ノスタルジア」,「プリンス・オブ・ペルシャ」,「栄冠は君に」。──編集部, LOGIN, 3号, 16-23pp.
▶Software Review
シューティングゲーム「ソル・フィース」を解説。──X68000新聞社, LOGIN, 3号, 200-201pp.
▶X68000新聞
「ファランクス」「ブルトン・レイ」と野球ゲーム特集「栄冠は君に」「生中継68」「ワールドスタジアム」の紹介。タケルでも販売が開始された「電脳倶楽部」など。──編集部, LOGIN, 3号, 252-255pp.
▶世界のパソコンシリーズ第1回 X68000の巻（後編）
前回のペーパークラフトX68000の続編。──編集部, LOGIN, 3号, 綴じ込み付録
▶NEW SOFT
「ノスタルジア」と「Misty Vol.7」を紹介。──編集部, LOGIN, 4号, 25, 29p.
▶最新ゲーム徹底解剖!!
「続ダンジョン・マスター カオスの逆襲」を攻略。──編集部, LOGIN, 4号, 130-133pp.
▶Software Review
「アトミック・ロボキッド」を紹介。──K. サワノフ, LOGIN, 4号, 160-161pp.
▶X68000新聞
メタボールのレンダリングが可能になった「C-TRACE＋」, 開発中の「MultiwordPRO-68K」「Musicstudio PRO-68K Ver.2.0」「ファランクス」「ノスタルジア」を紹介。──編集部, LOGIN, 4号, 212-215pp.
▶ゲームレビュー
「ニューラル・ギア」のゲームレビュー。──編集部, マイコン, 3月号, 198-199pp.
▶X68000マシン語入門
対話型ソフトを作る, の2回目。画面処理を大幅に強化, 住所録プログラム風の画面になった。──高橋雄一, マイコン, 3月号, 328-334pp.
▶なんでもQ＆A
Musicstudio PRO-68Kのバージョンアップについて答える。ほかに日本語FEP, ASKのメイン辞書とサブ辞書との違いについて。──シャープ液晶映像システム事業部第2商品企画部, マイコン, 3月号, 400-401pp.
▶フリーソフトウェアの世界
フリーソフトウェアの世界を法律や実際のソフトを交えて紹介する。フリーソフトウェアとPDSの違いの説明など。──編集部, ASCII, 3月号, 314p.
▶FREE SOFTWARE INDEX
大手ネットワークにアップロードされたPDSのうち, ここ1〜2カ月で目立ったものをピックアップ。テキストファイル縮小印刷ユーティリティ, SRAMディスクドライバなど。──編集部, ASCII, 3月号, 400-402pp.
▶お楽しみディスクについて
付録ディスクの解凍についての説明。X68000用はスプライトテストプログラムと1990年の索引。──編集部, ASCII, 3月号, 433-434pp.
▶GAME BOX
「イメージファイト」のゲームレビュー。──市原昌文, I/O, 3月号, 115p.
▶家計簿 for X68000
PC-9801/J3100版家計簿Ver3.62をX68000用に移植。ほとんどのコマンドがマウス対応になっている。2Mバイトのメモリがあればコンパイル可能で, 一気に高速化できる。プログラムは添付ディスクに収録。──Hm, I/O, 3月号, 198-199pp.

ポケコン

PC-E500
▶BREAK BLOCK
同じブロックを重ねると消える。すべてのブロックを消せば面クリア。ブロックパズルゲーム。──小川章, マイコンBASIC Magazine, 3月号, 167p.
▶ガンバレセイビイン3 〜続タカラヲ ミツケロ〜
セイビインを×印のところまで移動させる。ただし1歩進むごとに邪魔なブロックが出現するぞ。──町野稔, マイコンBASIC Magazine, 3月号, 168-170pp.
▶ポケコンQ＆A
LCDコントローラを直接操作してキャラクタを表示する。編集部, ポケコンジャーナル, 3月号, 66p.
▶DROP BLOCKS
パズルゲーム。全10面である。──伊藤正宏, ポケコンジャーナル, 3月号, 83-85pp.
▶アクセク アクセス
PC-E500シリーズ用のデータベースプログラム。名前, 住所, TEL, 会社名を入力できる。──わちがねどぅ, ポケコンジャーナル, 3月号, 93-95pp.

ポスト情報社会の到来
表紙の写真がうさんくささを際だたせているが, それはこの際置いておこう。すでにポスト情報社会をさぐる時代になっているらしい。情報社会が思考を援助したり混乱させたりする社会であれば, その次は精神に向かうだろうとのことだ。そういった社会構造の変革を7つの技術革新を例に語っている。クリスタルからアモルファス, エレクトロニカルからオプティカルといった具合だ。はたして, 精神の飛躍はなされるだろうか。 (K)
月尾嘉男著 PHP研究所刊 ☎03(3239)6261
四六版 215ページ 1,300円

ニューロコンピューター
AI→ファジィ→ニューロ, というわけでもないだろうが, ニューロという名前はけっこう有名になった。が, それがどういうものかをイメージできる者は少ない。本書はそういった人向けに平易にニューロコンピュータを解説した本である。面倒な数式も固有名詞も避けながら, ニューロコンピュータの歴史から連想記憶モデル, そしてニューロチップの話まで多岐にわたって語られるので, 誰にでも勧められる読み物となっている。 (K)
甘利俊一著 読売新聞社刊 四六版 244ページ
1,300円

書籍の価格は消費税込みです

QUESTION and ANSWER

X-BASICでHOME関数やBG-SCROLLで，グラフィックやBGをスクロールさせると絵がぶれてしまいます。以前，垂直同期がなんたらかんたらということを聞きました。絵をぶれないようにスムースにスクロールさせることはできないでしょうか？（Cまたはアセンブラでもよい）詳しくお願いします。

東京都　小川　敦

X68000の画面表示にはテキスト/グラフィック画面を管理するCRTCとスプライトコントローラ，そしてそれらを有機的に結合するビデオコントローラがかかわってきます。

グラフィック画面のスクロールとBGのスクロールは扱うコントローラが違うのですが，ここではグラフィック中心に解説します。BGについてはCRTCをスプライトコントローラに置き換えて読んでください（アドレスなどは当然異なります）。

通常のCRTディスプレイは左上から右下まで電子銃から放たれた1本の走査線を走らせることで画像を表示します。これは垂直表示期間－垂直帰線期間－垂直表示期間……と，表示期間と帰線期間を交互に繰り返しています。画面の左上から右下に向かって，画面の描き替えをしている期間が垂直表示期間です。画像がぶれてしまうのは，垂直表示期間にBGやグラフィックをスクロールさせている，つまり画面の上半分を描き終わっているのに下半分を別の画面を描いていることになるからです。ですから，ぶれを防ぐには，垂直帰線期間にCRTCに値を書き込むようにするのです。CRTCとはCRTコントローラの意味で，主に画面を作る走査線がきちんと構成されるように同期をとる役目をします。

まず，垂直帰線期間を調べる方法ですが，2つ方法があります。ひとつはCRTC割り込みを使う方法です。X68000にはたくさんの割り込みが用意されていますが，IOCSコールを使えば割り込みアドレスなど，割り込みに必要な最低限の情報をレジスタに設定してやるだけで，難しい処理はすべて勝手にやってくれるので簡単に割り込みを設定することができます。

これは便利ということで，作ったのがIOCSコール_VDISPSTを使ったサンプルです（リスト1）。エディタからプログラムを入力して，アセンブル，リンクして実行ファイルを作成します。プログラムを実行する前に，画面モードが768×512でグラフィック画面がRAMディスクなどに使われていないことを確認してください。また，すでに垂直同期割り込みが使われている場合にもうまく動作しません。

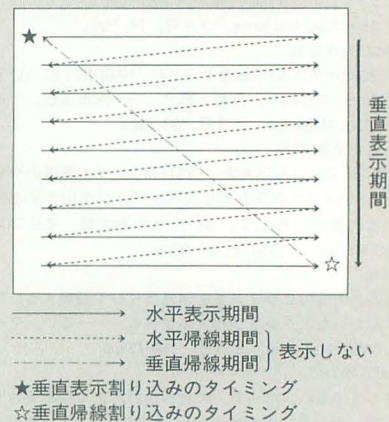
プログラムは割り込みがかかると，グラフィック画面を8ドットスクロールさせるようになっています。実行すると画面にボックスが表示され，右から左へボックスがスクロールします。しかし表示されたボックスが歪んでますよね。これは垂直表示期間に割り込みがかかるように設定しているからです（58行）。一度リセットして（割り込みを解除するにはリセットするしかない。サンプルなので手抜きです）58行を削り，56行の＊を取って，もう一度実行ファイルを作って実行してみてください。今度はちゃんとボックスが表示されました。いうまでもないことですが，垂直帰線期間に割り込むように変更したからです。

ついでに割り込み周期の設定について少

図1　走査線のようす

●リスト1

```
 1: i*
 2: * scroll sample program test1
 3: *
 4:
 5: _KEEPPR         equ     $ff31
 6:
 7: _VDISPST:       equ     $6C
 8: _G_CLR_ON:      equ     $90
 9: _HOME:          equ     $B3
10: _BOX:           equ     $B9
11:
12: *
13: * 入力
14: *       D0.L    $6C IOCSコール番号
15: *       A1.L    割り込み処理アドレス
16: *       D1.W    期間×256+カウンタ
17: * 戻り値
18: *       D0.L=0  割り込みが設定された
19: *           ≠0  割り込みが使われていた
20: *
21: * 期間:0          垂直帰線期間で割り込み
22: *       1          垂直表示期間で割り込み
23: *
24: * カウンタ:      0~255 (0は256として扱う)
25: *
26:         .text
27:         .even
28:
29: scroll:
30:         movem.l d0-d7/a0-a6,-(sp)
31:
32:         moveq.l #_HOME,d0
33:         moveq.l #0,d1                   * ページ0
34:         move.w  x,d2                    * x
35:         moveq.l #0,d3                   * y
36:         trap    #15                     * スクロール
37:         addq.w  #8,d2                   * x=x+8
38:         andi.w  #$3ff,d2                * x=x and 1023
39:         move.w  d2,x
40:
41:         movem.l (sp)+,d0-d7/a0-a6
42:         rte
43: x:
44:         dc.w    0
45:
46: sample:
47:         moveq.l #_G_CLR_ON,d0
48:         trap    #15
49:
50:         lea.l   param,a1
51:         moveq.l #_BOX,d0
52:         trap    #15
53:
54:         moveq   #_VDISPST,d0
55:         lea.l   scroll,a1
56: *       move.w  #$00_01,d1              * 垂直同期期間
57:
58:         move.w  #$01_01,d1              * 垂直表示期間
59:         trap    #15
60:
61:         clr.w   -(sp)
62:         move.l  #sample-scroll,d0
63:         move.l  d0,-(sp)
64:         dc.w    _KEEPPR                 * 常駐終了
65:
66: param:
67:         dc.w    0
68:         dc.w    0
69:         dc.w    767
70:         dc.w    511
71:         dc.w    15
72:         dc.w    $ffff
73:
74:         .end    sample
75:
```

●リスト2

```
 1: *
 2: * vdisp.fnc
 3: *
 4:
 5: GPIP:           equ     $e88001
 6:
 7: _B_BPEEK:       equ     $82
 8:
 9: * infomation table
10:
11:         .dc.l   x_init
12:         .dc.l   x_run
13:         .dc.l   x_end
14:         .dc.l   x_sys
15:         .dc.l   x_brk
16:         .dc.l   x_ctrl_d
17:         .dc.l   x_res1
18:         .dc.l   x_res2
19:         .dc.l   ptr_token
20:         .dc.l   ptr_param
21:         .dc.l   ptr_exec
22:         .dc.l   0,0,0,0,0
23:
24: x_init:
25: x_run:
26: x_end:
27: x_sys:
28: x_brk:
29: x_ctrl_d:
30: x_res1:
31: x_res2:
32:         rts
33:
34: ptr_token:
35:         dc.b    'vdisp',0
36:         dc.b    0
37:
38:         .even
39:
40: ptr_param:
41:         dc.l    vdisp_par
42: vdisp_par:
43:         dc.w    2
44:         dc.w    -1      * 戻り値なし
45: ptr_exec:
46:         .dc.l   vdisp
47:
48:         .text
49:         .even
50:
51: vdisp:
52:         tst.l   12(sp)
53:         bne     vdisp4
54: vdisp2:
55:         lea.l   GPIP,a1
56:         moveq.l #_B_BPEEK,d0
57:         trap    #15
58:         btst.l  #4,d0
59:         beq     vdisp2
60: vdisp3:
61:         lea.l   GPIP,a1
62:         moveq.l #_B_BPEEK,d0
63:         trap    #15
64:         btst.l  #4,d0
65:         bne     vdisp3
66:         clr.l   d0
67:         rts
68:
69: vdisp4:
70:         lea.l   GPIP,a1
71:         moveq.l #_B_BPEEK,d0
72:         trap    #15
73:         btst.l  #4,d0
74:         bne     vdisp4
75: vdisp5:
76:         lea.l   GPIP,a1
77:         moveq.l #_B_BPEEK,d0
78:         trap    #15
79:         btst.l  #4,d0
80:         beq     vdisp5
81:         clr.l   d0
82:         rts
83:
84:         .end
```

し触れておきましょう。割り込み周期は垂直同期周期×カウンタで決められます。垂直同期周期はディスプレイモードが31.5kHzのとき18.03msec，15.0kHzのとき16.27msecとなっています。ですから高解像度モードで，だいたい1秒ごとに割り込みをかけたかったら，1/0.01803=55.46，小数点第1位を四捨五入して55をカウンタにすれば，0.01803×55=0.99165秒ごとに割り込みをかけることができます。

これらの割り込みは，当然X-BASICから設定できるものではありません。しかし，ないものは作ってしまえのX-BASICですから，これから説明する直接I/Oを調べる方法を使って，垂直帰線を調べる関数を作ってみることにします。

X68000はメモリにI/Oを配置する，メモリ・マップド・I/O方式をとっています。手元にあった『X68000ベスト・プログラミング入門』（千葉憲昭著　技術評論社）で調べるとCRTCのV_DISP信号は＄E88001の第4ビットに出力されているようです。このビットが0で垂直帰線期間，1で垂直表示期間を表しています。

プログラムと呼べるほどたいしたものではないのですが，一応リスト2の説明をしておきます。ここで参照している$E88001は，スーパーバイザ領域なので，通常このアドレスをアクセスしようとするとバスエラーが発生してしまいます。スーパーバイザ領域をアクセスするには，スーパーバイザモードで実行すればいいのですが，このプログラムでは_B_BPEEKを使うことで解決しています。

次に垂直帰線期間の検出を例にとれば，すでに垂直帰線期間であればループを回し，垂直表示期間まで待つようにしています。こうしておいて垂直表示期間から垂直帰線期間に移った瞬間を検知するようにしています。二度手間のようですが，垂直帰線期間を最大に使うためにこうしているのです。ゲームなどでは割り込みを使わず，このような処理にすると高速になります。

このプログラムをアセンブル，リンクしてできた＊.Xファイルの拡張子を＊.FNCに変更してBASICのあるディレクトリに置き，BASIC.CNFに，

FUNC＝ファイル名

を追加してBASICを起動すると，新しい関数VDISPが追加されます。この関数は引数に0を与えると垂直帰線期間，0以外を与えると垂直表示期間を検出するまで後ろの文を実行しないようにするものです。この関数を使ったサンプル（リスト3）を紹介しますので参考にしてください。

（影山　裕昭）

●リスト3

```
 10 /* scroll sample X-BASIC
 20 /*
 30 int x=0
 40 screen 2,0,1,1
 50 /*
 60 box(0,0,767,511,15)
 70 /*
 80 vdisp(1)
 90 home(0,x,0)
100 x=x+8
110 x=x and 1023
120 goto 80
```

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を上げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので，電話番号も明記してくださいね。

宛先：〒108 東京都港区高輪2-19-13
NS高輪ビル
ソフトバンク株式会社出版部
「Oh! X質問箱」係

FROM READERS TO THE EDITOR

いろんな意味で待ちに待った春がやってきました。おもてはポカポカしていて気持ちいいし，陽気でちょっと変な人も出てくる。ああ，春はいいことだらけ。なにしろ，外で寝ても風邪なんかひかないもんなあ。よかったよかった。

◆当大学付属病院放射線科では患者予約処理にX68000を使っていた。さらに，研究室のほうでは画像処理や画像データベースの構築にX68000を利用しようとしている。メインとしてがんばっておられる先生は，バリバリのパワーユーザーで必要なソフトは独自開発している。PC-9801一辺倒の付属病院内にこのような頼もしい味方を発見し，あらためてX68000の可能性の大きさに自信を深めた今日この頃である。

大田　謙一郎(39)山口県

まさに「ないものは作れ」を地でいっているという感じですね。これからもがんばってほしいものです。

◆現在，福島県に住んでいるのですが，ここは実家の広島と違い，公害が少なく天体観察にはもってこいです。それで思い出したんですが，PC-9801やFM TOWNSには天文シミュレーションソフトがあるのに，X68000にはないようです。どこか出してくれないものでしょうか？

野中　裕一郎(21)福島県

ないものは作ろう。

◆シミュレーションプログラミング入門には期待していたのですが，信号機シミュレーションはかなりむずかしいですね。単一の信号交差点にしても，「1.(主道路)青－(同歩行者用)青－(従道路)赤－(同歩行者用)赤，2.青－点滅－赤－赤，3.黄－赤－青－青，4.すべて赤，5.赤－赤－青－青，6.赤－赤－青－点滅，7.赤－赤－黄－赤，8.すべて赤，1.へ戻る」となり，この主道路と従道路の交通量により時間を配分するのですが，前後左右の信号との兼ね合いも考えなければなりません。都市における交通管制では少なくとも100機以上の信号機についてパターンを決めないといけません。試行錯誤，または長年の勘みたいなものが必要でしょうね。

新穂　義久(33)宮崎県

宮崎県警にお勤めだそうで，現場の実感なんでしょうね。

◆正月は沖縄へ帰省。ゲゲッ，1989年分のOh!Xが行方不明。うーん，困った（空港の保管庫かもしれない)。しかし，1月の海でまさかバーベキューをするとは……。地元の人間にこんなことがしれたら笑いものにされてしまうけど，名古屋からやってきた人は喜んでくれた。うーん，たぶんあれでいい。

伊舎堂　盛行(18)愛知県

沖縄だと1月でも暖かいんでしょう。泳いでいる人はいませんでしたか？　なにしろ沖縄ですからねえ。

◆年末年始ブータンに飛んで，トンサという村のお祭りを見てきました。戦前の日本を（僕自身は見てないけど）思い出させる雰囲気に満ちあふれた国ですが，この国も日本と同じような伝統崩壊の道を歩んでいくのかと思うと，少し胸が痛みました。3月はパロという街のお祭り。ふたたびブータンに飛びます。

村井　裕弥(32)東京都

ブータンという名のキャラクターがいたなあ。おっと，ありゃブースカか。

◆クラスでスケベで有名な私が，同じクラスの女の子と一緒に学校から帰ることになった。「オレは絶対に手を出していない！」といっても信用してくれるやつが何人いることやら……。

幸　俊威(16)大阪府

涙を流して訴えれば，いつかは信用してもらえることでしょう。

◆私は少年ジャンプがきらいである。なぜなら，私はマイナーなものが大好きだからだ。誰もFIのことを知らないときはよかった。ひとりで暗く寂しく楽しんでいたのに。くそお。でも，いいんだ。だって，私にはまだスポックやMr.カトーがついているんだもん。いけいけ，エンタープライズ号。

福田　弘一(20)大阪府

「宇宙大作戦」はマイナーなんでしょうか？　結構メジャーだとも思えますんで，もっとマイナーなものを探してください。

◆この間，生まれてから2回目の金縛りにあった。1回目のときは夜中に起きたらなっていたという普通のものだったけど，今度はテレビを見ながらこたつで横になっていたときになった。うとうとしていたとき，気がつくと手足が動かなくなっていた。目はほんの少し開けたけど，ほとんどなにも見えなかった。ここまでは別に不思議なことはないけど，こたつが熱かったため，どうしても抜け出したくて力を入れたら，右手が少しずつ動いた。だけど，抜け出すには全然力が足りなくて無理だった。必死だったので内容は覚えていないけどテレビの音は聞こえた。腕が動いたのは本当だと思う。ほっぺたを押したのを覚えている。

小倉　修(18)埼玉県

金縛りには1回だけなったことがありますが，夢なのかどうかもおぼろげでした。

◆僕の住んでいる美原町には電車というものが走っていない。近くに近鉄と南海の2つの私鉄が走っているけど，どうやら美原町を避けているようである。それで，美原町は「陸の孤島」と呼ばれている。しかし，Oh!Xはちゃんと発売日には本屋に並んでいるところが，やっぱり大阪である。

燈田　安幸(18)大阪府

とはいえ，バスぐらいは通っているんでしょうね。

◆音楽プログラムの上達法をぜひ特集してほしいですね。もしくは「LIVE in '91」でボツ，あるいは採用一歩手前といった作品の「ここを直せばもっとよくなる」みたいなお手本を示してくれるコーナーなどがあれば，とても参考になると思うのですが。

井原　浩樹(25)東京都

まずはいい音楽をどんどん聞きましょう。

◀鈴川　美佳子　東京都
「なぜ68版WIZは出ない!?」の字のでかさに不満の大きさが現れていますね。たしかに「WizardryV」だけでも出してほしい。

◀小川　裕美　山口県
この方もすっかり常連になってしまいましたね。ちょっと絵のタッチが変わったような気もしますが，気のせいでしょうか。

▲上田　孝一　福岡県
背景がごちゃごちゃと書かれているのに対し，手前の女の子はわりとあっさりしていますね。まるで，水木しげるの絵のようだ（おおげさ）。

▲河野　純也　宮崎県
文面には「ボ，ボツにしないでくれ」と書かれています。何枚も送っていただいているようですが，この絵が特によかったですよ。

◆ちょっと遅くなったけど，僕の今年の抱負。
1）プログラムが作れるようになりたいけれども，まずはあせらずソースプログラムを入力したり，自分なりの説明をつけたリストを作るなど，“努力”を惜しまずにやりつづけたい。
2）我が愛機X68000EXPERTちゃんの環境整備をしたい。まず，4MバイトRAMボード，SCSIボード，80MバイトHD，拡張I/Oスロット，CZ-8NSI，数値演算プロセッサボード，C compiler PRO-68Kなど。
　今年1年かけて気合いを入れてやるぞ！
山内　富仁(23)北海道
　1番目は自分次第ですが，2番目は金次第ということになりそう。

◆2月号の表紙を見たとき，「悪魔くん千年王国」を思い出してしまいました。家獣のリアルバージョンですか。CGになってからでは，いちばんいい表紙だと思いました。
岡部　好充(17)愛知県
　やっぱりそう思う人がいましたか。ひと目見たときにそう思ったんですが，本当は何なんでしょうね。

◆結婚して，主人がパソコンをやっていたので私もやりはじめました。少しわからない面もありますが，面白いなあと思いはじめた，今日です。
伊藤　将子(24)愛知県
　だんだん泥沼にハマっていってください。

◆冬休みにやっと念願の内蔵RAMを買いました（僕のは初代X68000）。しかし，本体のカバーを外し，シールド用の鉄板を外すときに，ネジを落としてしまいました。そこで，僕は何を思ったのかX68000を振ったりしたのです。そのせいか，せっかくRAMを増設したのに，今度はディスクドライブのほうがおかしくなってしまいました。なんてバカなことをしてしまったんだあ。ちなみに「闇の血族」は個人的に好きです。
永井　徹(18)滋賀県
　振ったり，叩いたりというのは常套手段ですからね。でも，パソコンは……。

◆ドライブ1を壊してしまった。ノーブランドのフニャフニャしたディスクは使わないほうがいいと思う。ドライブから出てこないから引っ張ったら，機械的な部分が壊れた。よくわからんけど，白いプラスティックの部品でディスクと噛み合う部分だ。ついつい，ペンチで引っ張ってしまって，荒っぽかったかもしれない。
渡辺　篤志(18)滋賀県
　みんな危ないなあ。

◆ここしばらく，レポートが忙しくてまともに眠っていません。食後の睡魔がこわい。机の上でもJRの中でもなくて，ふとんの中でぐっすり眠りたい。しかし，私は今夜もまたオレンジ色のスイッチに手を伸ばし，さらに偏平な黒い物体で……。眠いよおう……，ぐしぐし。
岩瀬　貴代美(19)福岡県
　ちゃんと寝ないと美容に悪いですよ。

◆ラグーンのミュージックモードはディスク2を使わなくてもいいんです。ディスク3を4回入れたあと，システムディスクを入れればいいんです。
林　裕司(14)福島県
　読者の皆様にお手数をかけさせてしまいました。貴重なお時間をすいません（ここまでいうとイヤミか）。

◆2月号を買おうと発売日に本屋に行ったら，「少ししか仕入れていないので，定期予約の方のみです」と断られた。そして，別の本屋に行ったら同じことをいわれた。何軒か探したが見つからなかったので，名古屋のいとこに電話して送ってもらった。送られてきたOh!Xを見て，思わず涙ぐんでしまったのはいうまでもない。
服部　直幸(17)熊本県
　「涙でOh!Xが見えない」という感じ？　それはご迷惑をかけました。

◆2月号のTHE USER'S WORKSも12月号のグラディウスに勝るとも劣らないゲームが出てましたねえ。このゲーム，もしかしたら市販されているヘタなアドベンチャーゲームよりしっかり作られているんじゃないかなあ，って思うんですけど……。けど，グラディウス，このガイアの牙，そして次回もX1turboのゲーム。X1ユーザーの力はすごいですね。次回のゲームももしかしたらすごいゲームなんじゃなかろうか……。期待してます。
桑田　義久(17)千葉県
　本当にユーザーのパワーが実感できるコーナーになっています。

◆近所の本屋さんにはフロッピーディスクが置いてあります。「すごい店だ。時代の先端を走っている」と思いました。が，値札を見てみるとタイムスリップしたかのようです。なんと2HDが10枚で5,000円もするのです。ビックリしているところへ，中年のオジサンがやってきて買っていきました（実話）。
倉知　和弘(16)北海道
　しかし，いくらで仕入れているんだろう。すごい問屋があるのかな。

◆アクションロールプレイングゲームを作ろうと思っていますが，マシン語とC言語，どちらを先に覚えればいいのでしょうか？　プログラムをするには，どれを先に覚えればいいのですか。X-BASIC，マシン語，C言語。
猿渡　誠一(18)愛知県
　やっぱりABCの順でしょう。アセンブラ，BASIC，C言語。

◆私の通っていた小学校はボロだった。2階の床を破り1階の天井を貫通したり，2階の水道のお陰で雨漏りのように1階が水浸しになったり，鉛筆を落とすと床の隙間から床下に落ちて取れなくなったりした。だが，楽しかった。現在，その校舎はない。寂しいかぎりです。私は変わり者なのかもしれませんが，古い校舎のほうがよかったなあと思っています。
藤原　彰人(20)岡山県
　小学校はボロいほうが怪しい部屋とかがあって面白い。

◆「ちびまる子ちゃん」のまる子は正面から見ると鼻がない。しかし，横から見ると鼻はある。いったいどういう構造をしているのだろうか？
上野　政幸(17)京都府
　アトムの髪型と同じようなことじゃないでしょうか。どこから見ても角(?)は2本。

◆NHKスペシャル「電子立国日本の自叙伝」（全6部の予定）が放送されます。半導体や電卓戦争のことを取り上げるそうですが，最終部が9月放送予定というのが気になります。
大森　基弘(20)滋賀県
　間が空くと前の回の内容を忘れそう。

◆難しい特集に嘆息しながらページをめくっていると，いきなり「スプーンおばさん」の文字が。思わず中学時代を思い出してしまいました。好きな（「スプーンおばさん」が）女の子と一緒に歌ったりしたもんです。元の歌を歌っていたのは，あの飯島真理さんでしたよね。そうかあ，もう8年もたつのかあ。と，ちょっとの間，感傷に浸らせてくれたプログラムに感謝します（といっても，打ち込んで聞けないからちょっと悲しい）。
古澤　重治(20)岡山県
　そうかあ，もう8年もたつのかあ。

◆センター試験の御守りは，金沢神社の御守り（友人にもらった），某ラジオ番組のFellow's Card（2日目のみ），そして，電脳倶楽部Tシャツ（当たった）でした。結果は自己採点で721点。どえらい効き目。2次試験のときもよろしく。
寺門　修司(19)兵庫県
　Tシャツだけだとカンニングと間違われそう。うえになんか着ていたとは思いますが。

◆聞いた話によると，どこかの会社ではシムシ

ティーが教育ソフトになっているとか。決められた敷地内にいかに効率よく建物を配置するかということです(もちろん美しく)。たしか，設計会社だと思ったけどなんていうところかは忘れてしまった。　　宮沢　弘俊(18)長野県

そんな教育ソフトなら大歓迎。遊んでいるみたいなものですからね。

◆2月8日，私は防衛大学に合格しました。でも，その日のとある新聞の夕刊には，「入学定員500人に対し，合格者は前年より54人多い1344人という水増しぶり」と鬼の首をとったかのように書いているのを読んで，なにか値打ちを落とされたようで腹が立ちました。

　　舛井　淳祥(18)和歌山県

なんでも鬼の首をとったかのように書きますから，気にしない気にしない。

◆X1用の究極の連射回路を作りました。ジョイスティックポートからデータを読むたびにトリガを切り替えるので，常に最高速が得られます。これはPCエンジンのパッドを見て思いつきました。PCエンジンはすごい。さすがハドソン。

　　小谷　史樹(21)北海道

連射の高橋名人はいまどこに？　まあ，もともとハドソンの社員なんでしょうけど。

◆電脳倶楽部のTシャツを友人たちに見せたところ，「怪しい，怪しい」と立派な褒め言葉が返ってきて，自分としてはとてもウハウハしております。風邪の季節ですが，編集部の皆さんも体調に気をつけてがんばってくださいませ。

　　藤田　康一(19)静岡県

風邪なんかひきませんよ。なにしろ，○○ですから。

◆Z's-EX，ありがたく使わせていただいております。私もX68000と付き合いはじめて，そろそろ3年になるのでここらでプログラムを投稿するのもいいかななどと考えていたのですが，Oh!Xに採用されているのはどれも私の能力をはるかに上回っているので，もう少し修業を積んでからにしようかな，と弱気になっています。周りにプログラミングの好きな人が少ないのが残念なかぎりです。　　武藤　一文(18)埼玉県

修業を積むと聞くとインドで滝に打たれている姿を思い浮かべてしまう。

◆ふと気づくといまは2月10日。受験だからアンケートハガキを書くのを忘れていたのです。先日，某工業大学の合格発表がありました。新聞を見ると自分の名前は載っていない。ああ，浪人かと思っていたら，通知には“補欠”だって。判決が2週間伸びました。

　　木村　幸生(18)愛知県

結果はいかに？

◆小学生のころ，兄がPC-6001mkIIを買ったときからか，それとも自分がX1turboを買ったときからか。どちらから自分のパソコンに対する付き合いが始まったのだろうか，と思いながら，いつもアンケートハガキのパソコン歴の欄を書いています。1年ほど前に掘り出されたPC-6001mkIIをもらい，先日友人から壊れた（ただ，文字が出ないだけで正常に動く）X1Ckを譲ってもらった。変なやつと思われるかもしれないけれど，捨てるのはもったいないと思ったからだ。PC-6001mkIIとX1Ckは自分にとって宝であり，8ビット時代の遺産である（たとえ使われなくても)。X1turboに関してはいまだ現役で働いている(まあ，当然のことだけれども)。皆さん，8ビットマシンを大切にしましょう。8ビットがあったからこそ，いまの16/32ビットがあるのですから……。　永井　雅晴(19)愛知県

「たとえ使われなくても」なんていわずに，機会があれば使ってあげてください。

◆増設RAMを買う予定だったのに，車の事故でだめになってしまった。アコードのバンパーに本当に少しキズがついただけなのに，バンパー交換で4万円の出費でした。悲しい。

　　佐藤　秀昭(20)福井県

バンパーをコンニャクに変えましょう。

◆先生の目を盗んで数カ月。やっと完成した対戦型コラムス。コンピュータ実習中，殺気(!)がないときにしか遊べないけど，楽しいよお！　作ってよかった。えっ，来年はこのIBMのパソコン使えないの？　ゲッ，ちょっとショック……。

　　小嶋　健太朗(19)愛知県

後輩への置き土産にしてあげれば？

◆昨年の2月号からOh!Xを買いはじめた。買うようになったきっかけは，表紙に圧倒されたからだ。見ていくうちに興味深くなりパソコンがほしくなって，6月についに買ってもらったのだ。これからもがんばってください（まったく話がまとまらなかった)。

　　土居　正己(16)愛媛県

せっかく買ってもらったんだから，使いまくってあげましょう。

◆ゲーマーからプログラマへの転身を宣言した次の日，マウスの左ボタンがいかれてしまった。そして，3日後には自作ジョイスティックのGNDが断線した。これは神の思し召しだと思って修理をしていない。が，ファミコンを借りてドラゴンクエストIVをしている。いつになったらプログラマになれるのだろうか。

　　淮田　一磨(18)兵庫県

飽きたら，じゃないですか。

◆情報社会って「いかに必要な情報を得るか？」ではなく，「いかに不必要な情報を排除するか？」だったんですね。使いもしないのにダウンロードしまくったPDSに占拠されたハードディスクを整理しながらつくづくそう思います。

　　野田　敏之(19)神奈川県

それはよくいわれる言葉ですが，身をもって感じてしまったわけですね。

◆いま，僕たちの部活ではコンクールに向けてカードゲームを作っています。そこで重要なのがカードの絵の部分です。ほかにも絵を作っていますが，まず，カードのところの絵が下手かどうかで違ってきます。だから，今月のCGの特集をやってくれて，とてもためになりました。これから忙しいですが，がんばっていこうと思います。　　武藤　信行(13)愛知県

ゲームの内容も肝心なところですから，がんばってください。

◆最近，なぜかシューティングゲームにのめりこめない。ひと昔前は何回やっても「次こそクリアしてやる」と思えたのに……。年をくってしまったのかもしれない。心を貫くシューティングゲームがやりたい。

　　田中　剛一郎(18)東京都

心臓を貫くシューティングゲームを！（アブナイ）

◆いま，C言語で2人用ポーカーゲームを作ろうと思っています。2月号のスロットポーカーは大変参考になりました(特に役の判定)。単にポーカーを作るのはつまらないので，みんながアッと驚くようなルールをつけようと思っています。期待してください。

　　高橋　直之(20)静岡県

そのルールとはひょっとして，「足して21とか9になったら勝ち」とか。

◆ついにハードディスクを買いました。冬休みのバイト代がふっとびましたが，さすがに80Mバイトは広いですね。まだまだ余裕たっぷりです。話は変わりますが，去年の6月号，今年の1月号とディスクが付録になった月に，友人をひとりずつOh!Xの読者にすることに成功しました。機会があれば，また誰かを紹介したいと思います。それから，1月号のディスクのKLONDIKEは難しいですね。まだ，一度しかあがったこ

▲松本　浩幸　大阪府
そばかすの似合う女の子。かわいいですよね。素朴な感じに見えるからでしょうか。近頃では少なくなってきたかもしれない。

▲金子　聡　新潟県
最近，4コマ（これは3コマ？）漫画形式のものがあまり来ませんが，そちらも大歓迎しますので，みなさんよろしく。

とがありません。　　　榎　和彦(19)愛知県

KLONDIKEは何回もあがりました。もっとも，やった回数も何百回なんですが……。

◆大人のためのX68000，毎月楽しみにしています。予告の「確定申告にX68000を使う」，待ってましたという感じです。X68000はそのために買いました。いいソフトがないので，いまは半分子供のゲーム機と化しています。大人のためにページを増やしてがんばってください。

荒木　昇三(45)神奈川県

予告は跡形もなく破壊されてしまいました。どうもすみません。

◆このアンケートハガキはシムシティーをロードしながら書いている。おっと，終わったようだ。あ，だんだん呼び出し音が変わる。「ブーッブーッ」から「パァーパァーッ」となっているぞ。そんなにプレイしてほしいのか。よし，今夜も徹夜だ。　　山根　大樹(14)島根県

ロードしたのが運のつき。

◆今度，八ヶ岳に登ります。3月24日に出発しますので，そのあたりに登りに行く人は電脳倶楽部の豆しぼりを見かけたら声をかけてください。新聞には載らないようがんばります。しかし，金がかかる，かかる。山はパソコン以上に金をくう。　　永田　秀史(18)愛知県

山（まだ雪が残っているかも）は危険と隣り合わせですから，気をつけてください。

◆2月号の高橋さん，私の家はもっとすごいです。西にJR，北に旧国道という環境にある私の家は1日中貨物列車の振動があり，ひどいときには震度3ぐらい（マジ！）の揺れがあります。おまけに冬になると，なにかと電気を使ったりするので，よくブレーカーが落ちてしまうこともあり，気が気でありません。しかし，私のX68000ACE-HDは元気に動いています。

大塚　京吾(21)岐阜県

いままで運がよかっただけ？

◆女性読者が増えてきてなかなかよろしい。なかでも年間モニタの安井百合江さん。「なるほど，女性はこう見ているのか」と感心させられることがたくさんあるのです。いやあ，もっともっと増えてくれるといいですね。んでもって，

▲上田　修　岐阜県
ソフトバンク，Oh!XのA列車での旅立ち。どこへ連れていかれるのだろうか。きっと，とんでもないところだろう。

女性ライター登場！　なんてなるといいなあ。

石田　晴幸(18)千葉県

世の中には女性が半数いるんですから，もっともっとOh!Xの読者にも増えて然るべきです。どんどん増やしましょう。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

仲　間

★サークル「大魔界」では，第30回日本SF<i-CON>の自主企画としてライブRPGを行います。現在，スタッフ，プレイヤーなどゲームに協力していただける方を募集しています。あなたも真夏の夜の星の下，屋外プールサイドを利用したライブRPGに参加してみませんか。

日時　平成3年7月26日(金)　夜
会場　ルネスかなざわ

なお，ゲームに参加される方は日本SF大会への参加が必要条件ですので，連絡先を明記した紙片と72円切手を封書に同封のうえ，下記まで案内書をご請求ください。

〒920　石川県金沢市芳斉2-5-35　日交観トラベル金沢支店　第30回日本SF大会<i-CON>係

また，このゲームについてのお問い合わせは封書に連絡先を記入した紙片と62円切手を同封のうえ，下記までお問い合わせください。

〒933-03　富山県高岡市石堤540　尾崎　浩和（なお，封筒の表に「ライブRPG問い合わせ」とご記入ください）

★このたび，X68000のサークル“T.C.M.N.”を始めるにあたり，会員を募集します。主な活動としてMIDI，OPM，絵などを満載したディスクマガジンの発行を予定しています。初心者大歓迎。詳しくは62円切手同封のうえ，下記までお知らせください。〒995　山形県村山市楯岡中町5-12-201　設楽　亮(19)

★S-OSクラブ「Illegal」は4月より活動を再開する予定です。つきましては，4月より入会される方を募集します。主な活動としては，月1回の会報の発行，および，プログラムの作成，交換などを行います。S-OSに興味のある方なら初心者，エキスパートを問わず大歓迎です。連絡は年齢，電話，使用機種，S-OS歴を明記のうえ往復ハガキで下記まで。〒064　北海道札幌市中央区南20条西8-1-37　桑の実マンション1号室　渡辺　裕之(18)

★我らがサークル，SRKではゲームが大好きなX68000，およびPC-8801SR以降のユーザーを大募集しております。ゲーム好きにはたまらない内容が盛り沢山です。詳しく知りたい方は無料で会報ディスクを送りますので，住所，氏名，機種，電話番号を記入のうえ，下記の住所までお送りください。〒891-01　鹿児島県鹿児島市小原町17-6　蕪木　久夫(17)

売ります

★ポケコン「PC-1600K」，プログラムモジュール「CE-1600M」，文節変換辞書「CE-1650M」を合計45,000円（送料込み）で。連絡は往復ハガキでお願いします。〒854　長崎県諫早市船越町1182-3　菅原　実(38)

★「CZ-503F」（完動品，付属品，箱あり）を2個21,000円で（送料込み）。連絡は往復ハガキで。〒366　埼玉県深谷市藤ノ木65-1　加藤　勲(20)

買います

★X1用FM音源ボード「CZ-8BS1」を8,000円前後（送料込み）で買います。また，NECのプリンタ「PC-PR102TL3」（箱，マニュアル，ケーブル，付属品つき，ほとんど未使用のもの）を39,000円から42,500円（送料込み）で売ります。連絡はハガキで。〒349-13　栃木県下都賀郡藤岡町大字藤岡512-3　野代　広司(14)

★シャープのMIDIボード「CZ-6BM1」をマニュアル，ケーブル，付属品，箱つき，新品同様のものを9,000円から11,800円で買います。連絡は往復ハガキで。〒596　大阪府岸和田市加守町1-6-20　西村　貴明(22)

バックナンバー

★Oh!Xの1990年1～8月号を送料込み各1,200円で譲ってください。全部揃っていなくても結構です。連絡はハガキで。〒326　栃木県足利市大前町978-6　中村　伸之(15)

★Oh!MZの1984年6，11月号を送料込み各1,500円で買います。切り抜き不可。連絡は往復ハガキで。気長に待ちます。〒355　埼玉県東松山市材木町6-14　杉田　浩之(17)

★Oh!Xの1989年12月号，および1990年1月号を送料込み各1,500円で。ただし，広告以外のところの切り抜きは不可。少々の汚れなら可。連絡はハガキでお願いします。〒751　山口県下関市秋根上町589-1　笠井　透(17)

★Oh!Xの1987年1，9月号，および1989年2～7月号を送料こちらもちで各1,000円で買います。連絡はハガキでお願いします。〒382　長野県須坂市須坂1230-43-706　阿川　良輔(22)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は2月号の内容に関するレポートです。

●「HASH」の解説，どうもありがとうございました（といいつつもまったくわからない）。昨年の秋ごろ，ある書店のコンピュータ関係の書籍のところをうろついていたら，「CGレイトレ物語」なる本を見つけ，面白そうだったのでついつい買っちゃったんです。ひととおり読んでもよく意味がわかりませんでしたが，HASHの記事を読んでいたら「参考文献」として載っているではないですか。ということで，HASHの記事を読むときは手放せません。「参考文献」欄をこんなにありがたく思ったことはありませんでしたよ。とにかく，HASHで計算された絵を見て「影」というものの存在の重要さを思いしらされました。どんどん改善されていくといいと思いますね。カタいイメージのするレイトレーシングではなく，柔らかい影で柔らかいレイトレーシングへと変わっていくといいなあと思います。メタボールもいいですが，「影」のほうが先だと思いますね。CGをこよなく愛する者として，HASHを賞賛したいと思います。

ところで，友人がSX-WINDOW上で動くアセンブラや，ウィンドウが開きその中へオブジェクトをぱこぱこ放り込むとリンクしてくれるリンカ(そうとうメモリは食いそうだけど)があったらいいなともうしておりました。いかがなものでしょうか。私としてはもっともっと楽しめるものがほしいですね。ノート.Xの拡張版とか。これは彼もいっていたことですが，グラフィックエディタとか（簡単なものでよい）。

安井　百合江(16)　X68000 PRO　愛知県

●付録ディスクがついた直後の号でしたが，特にパワー不足は感じませんでした。たしかに2月号の記事は「先月号の付録の説明」が多かったですが，僕にとってその内容は知りたかったことや，知らなかったことなので，興味深く読めたのです。つまり，読者の求めるところ(知りたいところ，知らないところ)と記事の内容が一致すれば，ものたりなさを感じさせないものとなるのではないでしょうか。とはいえ，それがいちばんてっとりばやく，かつ，難しいのですが。

高橋　毅(19)　X68000 PRO,MSX2　埼玉県

●「マシン語カクテル」，回を追うごとに，なんだか学校のコンピュータの授業に近づいているようです。結構，知らなかったことが多くあるので助かってます。もっと，こういうのもやってほしいです。また，できればあまり特定の機種に限定しないでやってほしいです。

横山　賀一(18)　MZ-2500，PC-8801mkIIFR　東京都

●ハードウェア工作入門，とってもためになりました。実は本文中のLM358は私の電子工作歴6年の中で初めて“はじけた！”部品なのです。PC-9801で外部I/Oの基板を作り，温度センサの信号を4558で増幅し，ADC8089(あれ？　0809だったかな。とにかくこんな感じのADコンバータ)でAD変換後，8255を通してPC-9801に入力するという基板で，電源部(だったと思う)あたりにLM358があって，電源を入れた瞬間に「ぴきっ！」という音とともにあたりに悪臭が漂い。358のど真ん中に穴が空きました（2年前のことで回路をよく覚えていない。書いたことには間違いがあるかもしれない)。皆さん，電源は＋，－を正しくつなぎましょう。

畑　剛志(19)　X1turboZII/model 10，MSX/2，JR-100　北海道

●「Z's-EX」について。まず，前半のZ's STAFF用のアドインソフト（どっちが親といわれれば困るが）の作り方の部分がうれしい。これさえあれば，別にソースがなくてもZ's-EXにさらなる特殊効果を加えたりもできますしね。後半の個々のプログラムの解説もかなりわかりやすかった。

高村　信(20)　X1turbo,PC-8001mkII　東京都

●そうです，シャープ系のマシンでは，こんなにたくさんゲームがリリースされたことは過去にはないのです。したがって，いまが旬のX68000をかかえる我々としては，この企画に気合いが入ってもそれはきわめて自然なことであるわけです。きっと皆さんも，いままで歯がゆい思いをされてきたことでしょう。図表など，サマになるくらいの数と質が揃ったのも今年ぐらいでしょうね。おめでとうございます。

私は，毎年，あまりこの企画をしっかりとは見ていませんでした(スミマセン)。文字ばっかりで，なんだか読む気がそがれてしまっていたのです。しかし，今回は図表が2ページもあるので，そこから読みはじめて，結局全部読むはめになりました。図表は大正解ですね。大変面白いと思います。また，グラフの横に書かれているコメントには，笑わせていただきました。「カオスの逆襲」とか，「ひと夏の恋」とか，「光栄のファンは実に正直者揃いだ」とか，Oh!Xらしい毒舌（だと思う）は，さすがに時効ソフトならではというところでしょうか。

私は毎年，自由応募部門を楽しみにしていますので，今年も奇想天外なものを期待しております。“地球防衛軍のお姉さん”なんか，いま思い出しても笑ってしまいますよね。

浅野　憲(19)　X68000PRO，X1turboIII，X1Fmodel20，MZ-80C,FM-77L2，M5Jr.,PC-6001，PC-1245　大阪府

ごめんなさいのコーナー

3月号　ZMUSIC.FNC変更のお知らせ

P.55 文中に「……『網掛け』の部分の命令を削除してください」という記述がありますが，まったく判別できなくなっていました。矢印のところが該当部分です。

```
  tst.b    L000d67
  bmi      L000394
→sub.w    d5,L000e02
→bmi      L000394
  move.b   #$40,(a2)+
  move.b   #$4c,(a2)+
```

3月号　ライフゲームで姓名判断？

リスト8がリスト7と同じになっていました。詳細は今月号の72ページをご覧ください。もうしわけありませんでした。

バグに関するお問い合わせは
☎03(5488)1311(直通)
月〜金曜日16：00〜18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

協力スタッフ&年間モニタ待ってるからね

▼先月号でもお知らせしたとおり，次号5月号（4月18日発売）には3回目の付録ディスクがつくことになりました。ディスクの名称は「黄金週間PRO-68K」，値段は780円です。例によって限度を超えた盛り沢山の内容です。皆さん思いっきり期待してくださいね。

▼Oh!Xでは本誌の編集に協力してくれるスタッフを大募集いたします。仕事の内容は，原稿の執筆，プログラム開発，投稿作品のチェックなど多岐にわたりますが，時間的な拘束などはありません（ただし原稿には締め切りがありますが)。応募資格は東京近郊にお住まいの社会人および学生でOh!Xの誌面作りに参加したい人。希望者は，住所・氏名・年齢・電話番号を明記のうえ，自由原稿を5000字以内(本誌約2ページ分)にまとめ，Oh!X編集部「スタッフ希望」係までお送りください。お待ちしています。

▼Oh!Xでは，第7期の愛読者年間モニタの募集を行います。モニタというのは，1年間毎月Oh!Xをお送りし，いくつかの設問に答えていただくものです。モニタの皆さんから寄せられたご意見は私たちの雑誌の編集に役立てるとともに，そのレポートの一部は「DRIVE ON」のコーナーに掲載の予定です。応募方法は，住所・氏名・年齢・職業（学年)，使用機種などを明記のうえ，本誌へのご意見(レポート用紙2枚程度)を添えて，Oh!X編集部「愛読者年間モニタ」係まで郵送してください。採用者の発表は6月号で行い，7月号からレポートを依頼する予定です。年々モニタ希望者は減る傾向にありますが(昔のMZ/X1ユーザーに比べてX68000ユーザーはおとなしいですよ)，本誌の内容に関して意見や提案をぶつけてみたい方はぜひともご応募ください。

▼今月の「X68000マシン語プログラミング」は筆者の村田敏幸氏が多忙のため休載させていただきます。また「大人のためのX68000」は荻窪圭氏がスキーのためお休みさせていただきます。次号をお楽しみに。

投稿応募要領

- 原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒108 東京都港区高輪2-19-13 NS高輪ビル
ソフトバンク出版部
Oh!X「㋢㋮㋯㋱」係

S H I F T ・ B R E A K

▶今月もまたスキーに行ってしまいました。今度は体育の授業。顔をパンダにし，パラレルをマスターして帰って来ましたぜ。しかしGAME OF THE YEARが忙しい真っただ中に出かけたために，みなさまに多大な迷惑をおかけしてしまいました。編集諸氏，仕事を手伝ってくれた山田氏，金子氏，吉村氏，どうもありがとうございました。へこへこ。（浦）

▶「栄冠は君に」のレビューで，タッチアップができないと書いた。あとになって，あれは走者と3塁ベースコーチャーの走塁技術が低かったからだということがわかりました。自分の練習メニューの組み立ての粗雑さが露呈されてしまったわけで，なんとも恥ずかしい。ん，待てよ。するってえとタッチアップなしでベスト4に入ったのか。（H．K．）

▶ある日。ティラミスチョコなるものを買った男がいた。これはっ，うまさ大爆発ではないかーっ！　とゆーわけで翌朝，男はもういちどあの味をと，コンビニの前まで来て，はっと我にかえった。しまった。今日，男がチョコなんか買ったら気まずいぢゃないか。なぜならその日は2月14日，バレンタインデーだったのだ。（一日禁断症状だった男(で))

▶「ファイネスト・アワー」の最終面のボスの倒し方を誰か教えてくれー。ところで，来月のディスクのオープニング用（と勝手に決めていた）オリジナル曲は容量の関係でボシャリました。ラップとかそのへんのノリで作っていたらPCMデータが200Kバイトを超えてしまったのです。あと，未確認の予定ですが「Oh！X謎の別冊」が出るかも……。（善）

▶車を走らせているとき感じる接地感。これの正体はなんだろう。飛行機の浮遊感もそうだ。計算機で実現できるものなのか。グラフィックと効果音だけで表現できるものなのか。正確なシミュレーションを究めればよいのか。もっと賢い手段があるのか。ゲームというのは不思議で，ほんのわずかな違いで現実感が増すことがある。実に面白い。（A．T．）

▶4次元グラフィックやラジコンスティックなどユニークな記事を展開してきたK氏が先ごろご結婚とあいなりました。なれ染めはコンピュータとアマチュア無線だそうで，新居は高台，新婚旅行は無線機をかついで行かれるとか。なにはともあれ，おめでとうございます。古株のなかでは僕とKO氏が残ってしまったなあ。あ，社長のI氏がいたか。（S．S．）

▶さればこそ，ワールズエンドガーデンをきっかけに，避けて通っていたハウスミュージックに手を出してみた。してみると，なかなか気持ちがいい。かつて音楽にはメロディの呪縛があって，意味の呪縛があって，形態の呪縛があって，楽器の呪縛があった。ハウスはそれらすべてを破壊したのかもしれない。残るのはリズムだけだ（？），不明。（K）

▶ワープロで原稿を書いていると突然マウスの左ボタンが反応しなくなった。マウスなしでX68000付属のワープロを使うことの無力さを再認識してしまった。しかたがないのでマウスを買うことにしたが，秋葉原でさえ在庫のある店がほとんどない。歩き回ってやっと見つけたときには，これで原稿が落ちないですむという安堵感で満たされていた。（KO）

▶歯が痛いので歯医者に行った。昔，治療した歯に穴が空いてきたのである。こわくてなかなか行く気になれなかったが，放っておくとますますひどい状況になりそうなので，しかたなく通いだした。とりあえず，思ったより痛い思いはしていない。まあ，何事も決断のあとには（結果のよしあしにかかわらず）安堵が訪れるもの。行ってよかった。（A）

▶英会話やらバンドなぞ始めてしまったものだから，ますます眠る時間がなくなってきた。で，とうとうホテルにご宿泊。そうしてみると夜毎に飲みに行けたりするもんだから，ああ快適と思ってしまう。いかん，このままではなしくずしにホテルライフに突入だわ。でも，ま，それもいいかなと思い始めてたりして……。(いいDRUMMERいないかなのE．O．)

▶この春には32ビットは出ない，でもまあ，いずれ出てくるということだ。68020，コプロ標準装備，クロック40MHz，メインメモリ16Mバイト，LD-RAM搭載……仕様発表以来5年。発売予定日は刻一刻と近づいている。さてと，きっと今年最後のディスクだから，とりあえずあれとこれを入れて，あれ？　残り30Kバイト。どうしよう……。（U）

▶今年のアマチュアCGAコンテストは，ASAHIパソコン誌の後援もあって盛大であった。ただ，会場に来ていた人のほとんどがX68000ユーザーだったのが気になる。CGAシステムの98版が待たれるところだ。X68000ユーザーも他機種の作品に交じって評価を受けたら，いま以上にこのマシンでがんばってきたことを誇りに思えるだろうしね。（T）

microOdyssey

3月2日、第3回アマチュアCGAコンテストの発表会が行われた。本誌ではすっかりお馴染みになったCGAコンテストたが、今年はなんといっても銀座通りに面したYAMAHAホールという晴れ舞台、会場へ向かうパソコンユーザーたちの足取りも乱れようというものだ。「土曜日の午後に銀座を歩くなんてすごいことだよな」「そうですねぇ」と、私と（U）はツインビーの如く左右に大きく迂回したり交差したりしながら人の流れをかわしていく。しかし、隣のガスホールの前までくると、1枚の看板が目に飛び込んできた。「おたく展」。……今日は日が悪いんじゃないか。なにやら不吉な予感が脳裏をかすめたが……。

さてと、CGAは映像制作に臨む人にひとつの可能性を示した。個人もしくは少人数のグループがささやかな設備でアニメーションを制作する道を開いたからだ。それは同時に、パソコンユーザーの夢でもあった。

今回応募された作品の大半はDōGA CGAシステムを利用したものであり、そのポリゴン表示のグラフィックはそれ自体一定のイメージを形成する。だが、それをイメージの制約とするか、イメージの素材とするかは作家自身の持つ創造力の問題であろう。現状のCGAシステムで十分映像作品として評価されるものが作れることはすでに証明されたといってよい。あとはユーザーがどこまで作品を映像として高められるかということになる。

ただ、私たちパソコンユーザーにとってはもうひとつ問題があるように思われるかもしれない。ユーザーが作家になるための能力についてだ。さっき創造力といったが、よくいわれるセンスとか感性とか絵心とかいう抽象的な能力に関しては、世間では「才能」などといった理不尽な言葉で片づけられることが多い。

しかし、これらの能力のほとんどは理にかなった環境と努力と実績に裏付けられた技術だと私は信じている。どんなイメージも初めから作れる人はいない。だだ、気づいたときに差がついていただけのことなのだ。絵を描く能力は、筆を動かす、効果を見るというアクションとフィードバックの繰り返しによって形成されるものだ。それがパソコンを使ったシステムではこの繰り返しを高速に行える。CRTにはインクと用紙が無尽蔵に用意されているし、少々の失敗はものともしない。少しくらいの差であれば挽回のチャンスはあるはずだ。

そういった意味でもシステム側の課題は、ユーザーインタフェイスの向上にあるだろう。このことはユーザーの裾野を広げる意味でCGAシステムの使いやすさを向上させたいというDōGA代表の鎌田氏の見解とも一致していると思う。もちろんこれは作品を映像として高めるというのと同じくらい困難を伴う作業となるかもしれないが。

とまあ、強引にインタフェイスとCGAを結びつけたところで、お開きの位置まで進ませていただこう。そして不吉な予感は発表会の前半が終わったところで突如その姿を表した。休憩時間中にスクリーンに映っていたもの、いったいあれはなんだったのだろう……。その正体は？もしかしたら私自身の内面に潜むDARK SIDEだったのかもしれない。　（T）

1991年5月号4月18日(木)発売

特集　入門者のための環境構成術
第6回「言わせてくれなくちゃだワ」
ハード&ソフト春の新製品詳報
特別付録　黄金週間PRO-68K(5"2HD)
特別定価780円

バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(3233)3312
	〃	書泉ブックマートB1	03(3294)0011
	〃	書泉グランデ5F	03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(3257)2660
	八重洲	八重洲ブックセンター3F	03(3281)1811
	新宿	紀伊国屋書店本店	03(3354)0131
	高田馬場	未来堂書店	03(3200)9185
	渋谷	大盛堂書店	03(3463)0511
	池袋	リブロ池袋店	03(3981)0111
	〃	西武百貨店9F コンピュータ・フォーラム	03(3981)0111
神奈川	横浜	有隣堂横浜駅西口店	045(311)6265
	〃	有隣堂ルミネ店	045(453)0811
	藤沢	有隣堂藤沢店	0466(26)1411
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ5	0471(64)8551
	船橋	リブロ船橋店	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	0472(24)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコン∑上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある『新規』『継続』のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS（株）にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

Oh!X 4月号
■1991年4月1日発行　定価560円(本体544円)
■発行人　孫　正義
■編集人　橋本五郎
■発売元　ソフトバンク株式会社
■出版事業部　〒108 東京都港区高輪2-19-13　NS高輪ビル
Oh!X編集部　☎03(5488)1309
出版営業部　☎03(5488)1360 FAX 03(5488)1364
広告センター　☎03(3297)0181
■印　刷　凸版印刷株式会社

BACK ISSUES バックナンバー案内

ここには1990年4月号から1991年3月号までをご紹介しました。現在1990年11, 12, 1991年1～3月号までの在庫がございます。バックナンバーおよび定期購読のお申し込み方法については，180ページを参照してください。

1990

4月号(品切れ)
特集　ゲームシステム文学誌
1989年度GAME OF THE YEAR発表
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/C調言語講座/X68000マシン語
●X1・MZ-2000/2500用RPG The Cave of Dalk
●うわさの68040, ついに登場
LIVE in '90　バーニングフォース(OPMD対応)
THE SOFTOUCH The Fille Professor/HOST PRO-68K
全機種共通システム　ファジィコンピュータシミュレータI-MY

5月号(品切れ)
特集　BASICプログラミング
第5回　言わせてくれなくちゃだワ
連載　ショートプロぱーてぃ/Z80's Bar
X-BASIC調理実習/X68000マシン語プログラミング
●新機種X68000SUPER-HD/EXPERTII/PROII
●ラジコンスティックの製作
LIVE in '90　TURBO OUTRUN
THE SOFTOUCH　天下統一/ポピュラス/Hyperword
全機種共通システム　インタプリタ言語STACK

6月号(品切れ)
特集　創刊8周年記念PRO-68K(付録5"2HD)
Oh!Xアンケート結果大分析大会
連載　ショートプロぱーてぃ/Z80's Bar/PurePASCAL
X-BASIC調理実習/X68000マシン語プログラミング
●X1turbo用コマンドシェルシミュレータ
●ハードウェア工作入門
LIVE in '90　ナイトアームズ/悪魔城伝説/この木なんの木
THE SOFTOUCH　三国志II/FAR SIDE MOON/グラナダ
全機種共通システム　X68000用S-OS"SWORD"他

7月号(品切れ)
特集　マシン語への第一歩
X68000SUPER-HD試用レポート
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/PurePASCAL
●INTEGRAL XI――ノーマルX1への対応
●ハードウェア工作入門
LIVE in '90　夢幻戦士ヴァリスII/トッカータとフーガニ短調
THE SOFTOUCH　サーク/あーくしゅ/ダウンタウン熱血物語
全機種共通システム　リロケータブルアセンブラWZD

8月号(品切れ)
特集　ADVANCED 2D GRAPHICS
100号記念特別モニタプレゼント
連載　ショートプロぱーてぃ/Z80's Bar/INTEGRAL XI
X-BASIC調理実習/X68000マシン語プログラミング
PurePASCAL/ハードウェア工作入門
●X68000用画像回転プログラム XROT0.X
LIVE in '90　OMENS OF LOVE/ENDLESS RAIN/ダートフォックス
THE SOFTOUCH　大航海時代/ウルティマV/プロミストランド
全機種共通システム　リンカWLK

9月号(品切れ)
特集1 日本語を処理するための序章
特集2 ADVANCED 2D GRAPHICS
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
X-BASIC調理実習/マシン語プログラミング
PurePASCAL/ハードウェア工作入門
●清水和人流プログラミング道場
LIVE in '90　風の谷のナウシカ/ラジオ体操第一
THE SOFTOUCH　T&T/D-Again/シムシティー/ギャラガ'88ほか
全機種共通システム　BILLIARDS

10月号(品切れ)
特集　電子音楽術入門
連載　ショートプロぱーてぃ/Z80's Bar/DōGA・CGA
マシン語プログラミング/ハードウェア工作入門
清水和人流プログラミング道場
●荻窪圭の大人のためのX68000
●中森章のようこそここへC言語
LIVE in '90　Rise And Fall/PARADOX/キューピー3分クッキング
THE SOFTOUCH　ワールドコート/ルーンワース/闇の血族/提督の決断
全機種共通システム　ライブラリアンWLB

11月号
特集　理科系のGAME REVIEW
連載　Z80's Bar/DōGA・CGA/カードゲーム
マシン語プログラミング/ハードウェア工作入門
PurePASCAL/X-BASIC調理実習
ようこそここへC言語/INTEGRAL XI
荻窪圭の大人のためのX68000
LIVE in '90　ピラミッドソーサリアン/ザ・スキーム
THE SOFTOUCH SPECIAL　ラグーン/幻獣鬼/サイバリオン/GUNSHIP他
全機種共通システム　スクリーンエディタEDC-T

12月号
特集　XCのための傾向と対策
連載　X-BASICプログラミング調理実習/ハードウェア工作入門
マシン語プログラミング/ショートプロぱーてぃ/Z80's Bar
大人のためのX68000/ようこそここへC言語/INTEGRAL XI
●シミュレーションプログラミング入門
●特別企画アナログジョイスティックの製作
LIVE in '90　グラディウスIII/メタルサイト
THE SOFTOUCH SPECIAL　イメージファイト/ジェミニウイング/NAIOUS他
全機種共通システム　STACKコンパイラ

1991

1月号
特集　急接近! SX-WINDOW
特別付録　謹賀新年PRO-68K(5"2HD)
連載　ハードウェア工作入門/シミュレーションプログラミング入門
DōGA・CGA/ショートプロぱーてぃ/大人のためのX68000
PurePASCAL/清水和人流プログラミング道場/X-BASIC調理実習
LIVE in '91　めぞん一刻/涙で綴るパパへの手紙
THE SOFTOUCH　ソル・フィース/銀英伝II/続ダンジョン・マスター他
製品紹介　光磁気ディスクCZ-6MOI
全機種共通システム　ライブラリアンWLB

2月号
特集1　グラフィックの"実験的"手法
特集2　SX-WINDOWプログラミング
連載　ハードウェア工作入門/シミュレーションプログラミング入門
マシン語プログラミング/大人のためのX68000/Z80's Bar
ショートプロぱーてぃ/INTEGRAL XI/ようこそここへC言語
●1990年度 GAME OF THE YEAR ノミネート発表
LIVE in '91　Misty Blue/スプーンおばさん
THE SOFTOUCH　栄冠は君に/KLAX/ダイナマイト・デューク他
全機種共通システム　ダイスゲームKISMET

3月号
特集　MIDI & MUSIC PROCESSING
連載　ハードウェア工作入門/シミュレーションプログラミング入門
マシン語プログラミング/大人のためのX68000/Z80's Bar
ショートプロぱーてぃ/DōGA・CGA/C言語/PurePASCAL
●SXLIFE完結編/ウィンドウシステム大比較
●周辺機器新製品紹介
LIVE in '91　戦いの兜/LITTLE WING/リゾ・ラバ/花
THE SOFTOUCH　アトミック・ロボキッド/スペースローグ他
全機種共通システム　アクションゲームMUD BALLIN'

X68000 マシン語プログラミング

入門編

著●村田敏幸

B5判変型・388ページ　定価2800円(税込み)

ストロングスタイルのマシン語入門!!

●本書まえがきから

1990年11月現在。『Oh！X』誌に連載中の「X68000マシン語プログラミング」のうち<入門編>と称した冒頭部分を1冊にまとめたら，こんな本になった。マシン語プログラミングに興味をもったX68000ユーザーのための副読本，とでもいったらよいのだろうか。少なくとも，教科書的なプログラミング入門書では決してない。むしろ，問題集であり，実践テキストのノリに近い。

マシン語にかぎらず，プログラミングに関する知識/技術は，実際のプログラミングの中でこそ身につき，磨かれるものだ。この不変の真理にもとづき，本書は読者に自分の頭と体とを使うことを強いるように書かれている。エッセンスを100倍くらいに薄めて吸い差しでとろとろと流し込むような親切さは排除した。文書の襞に隠れた大小の謎は，サンプルプログラムを読み，動かし，改良することによって解き明かされるだろう。

●本書の内容

- CHAPTER 0　マシン語プログラミングの準備
- CHAPTER 1　マシン語プログラミングの流れ
- CHAPTER 2　68000の基本命令を覚えよう
- CHAPTER 3　12語の68000実習プログラミング
- CHAPTER 4　デバッガを使ってみよう
- CHAPTER 5　文字列操作の基本
- CHAPTER 6　正しいフィルタの作り方
- CHAPTER 7　コマンド作成“基本”作法
- CHAPTER 8　サブルーチンに汎用性を
- CHAPTER 9　「プロセス操作」という世界
- CHAPTER A　ファイル管理の方法
- CHAPTER B　デバイスドライバを作る
- CHAPTER C　脱“入門者”のための身辺整理
- APPENDIX　本書を読むための用語集
Human68kバージョンアップ履歴

X68000 マシン語プログラミング 入門編 村田敏幸

'91年3月末刊行予定

SX-WINDOW プログラミング

吉沢正敏●著

B5判変型　定価4500円(税込み)

X68000にイベントドリブン方式のマルチタスク・ウィンドウ環境を提供するSX-WINDOWは，X68000に新たな世界を拓くものとして熱い期待を集めている。本書は，このSX-WINDOW上でプログラムを作ってみたいと思っているユーザーを対象にした，プログラム作成のためのガイドブックである。イベントドリブン，リソースなどのウィンドウ・プログラムの基礎知識，サンプルプログラムによる具体例，ウィンドウ関連のシステムコール一覧など，SX-WINDOW上でプログラミングする際のエッセンスを集めている。

2枚のボードが1枚になった

KGB-X68PRK

写真はKGB-X68PRK-14です

広大なメモリ空間を実現する最大4Mバイトの
高速増設メモリ

高速演算を約束してくれる
数値演算プロセッサ

- メモリアクセスノーウェイトによる高速アクセス
- CZ-6BE2、CZ-6BE4、CZ-6BP1との混在が可能
- 複数枚のKGB-X68PRKの実装が可能
- ジャンパの変更により任意のアドレス空間にメモリの配置が可能
- ジャンパの変更により数値演算プロセッサの1枚目、2枚目、未使用の選択が可能
- 1M、2M、3Mメモリモデルは購入後もメモリ増設が可能
- PRK-10、11、12、13、14にはデバイスドライバ(FLOAT3.X)が付属

CZ-600C、601C、611C、652C、653C、662C、663Cで御使用の際にはあらかじめ専用の1Mメモリ(CZ-6BE1、A、B等)でメインメモリを2Mバイト以上にしておく必要があります。

製品価格一覧

製品	価格
KGB-X68PRK-00 (メモリ無し/数値演算プロセッサ無し)	¥34,000
KGB-X68PRK-01 (1Mメモリ/数値演算プロセッサ無し)	¥58,000
KGB-X68PRK-02 (2Mメモリ/数値演算プロセッサ無し)	¥74,000
KGB-X68PRK-03 (3Mメモリ/数値演算プロセッサ無し)	¥98,000
KGB-X68PRK-04 (4Mメモリ/数値演算プロセッサ無し)	¥122,000
KGB-X68PRK-10 (メモリ無し/数値演算プロセッサ付き)	¥76,000
KGB-X68PRK-11 (1Mメモリ/数値演算プロセッサ付き)	¥96,000
KGB-X68PRK-12 (2Mメモリ/数値演算プロセッサ付き)	¥112,000
KGB-X68PRK-13 (3Mメモリ/数値演算プロセッサ付き)	¥136,000
KGB-X68PRK-14 (4Mメモリ/数値演算プロセッサ付き)	¥160,000

購入後の増設費用

メモリ	
1Mバイト	¥24,000
2Mバイト	¥51,000
3Mバイト	¥76,000
数値演算プロセッサ MC68881RC16	¥38,000

PRK質問箱

、購入後のメモリ増設はどうやるのでしょう?
、ご購入後のPRKに対するメモリの増設は半田付け等の技術を要するため原則として当社に送り返していただき増設いたします。自分でメモリ増設をする場合は通信販売のみですが必要な部品の販売も致します。御希望の方はお問い合わせ下さい。

、数値演算プロセッサにMC68882を使用することは可能ですか?
、MC68882では動作しないソフトが存在するため使用できません。

、「数値演算プロセッサのみ」や「プロセッサ無しメモリ無し」のPRKがほしいのですが?
、PRK-10、PRK-00の型番で商品化しております。

最近PRKをスロットに挿入したが動作しないと言う御質問を良く受けますが、ほとんどの場合は差し込み不足が原因です。X68000のスロットは大変堅く裏蓋が閉まる状態でも差し込み不十分の場合があります。御注意ください。

おしらせ

バージョンアップサービス

★BASIC拡張関数パッケージ(B6-6306)
(C言語ライブラリー付き)
★C言語ライブラリー(B6-6305)
SHARP XC Ver.2に対応になりました。新バージョンでは従来のXFUNCLIB.Aの他に新たにXFUNCLIB.Lが追加されています。
★DISK CACHER
ハードディスクキャッシュの大幅な高速化が行なわれました。
HDISKCACHE.SYSのVer.2.00未満をお持ちの方が対照になります。

バージョンアップご希望の方は旧バージョンのディスクラベルと代金を同封して現金書留で通販部宛にお申し込みください。

B6-6306(拡張関数ライブラリー付き)	¥2000
B6-6305(C言語ライブラリー)	¥1500
B6-6304(ディスクキャッシャー)	¥1500

※送料、手数料、税込みの価格です。

充実のBASIC HOUSEソフトウェア&ハードウェア

速12BIT, 16CH A/Dコンバータボード(KGB-AD12) X1	¥118,000
ォトアイソレーション16BITデジタル入出力ボード(KGB-PIO) X1	¥ 42,000
イソレーション16BITデジタル入出力ボード(KGB-X68PIO) X68000	¥ 68,000
ンディプリンタ&インターフェース(HANDYPRINTjack) X68000	¥ 24,800
速12BIT, 4CH D/Aコンバータボード(KGB-DA4) X1	¥ 98,000
用ローコストA/D&PIOボード(KGB-X1S) X1	¥ 19,800
速12BIT, 16CH A/Dコンバータ(KGB-X68ADC) X68000	¥128,000
180CPUボードMach 180(KGB-CPXB) X68000	¥ 98,000
ーコストMIDIインターフェース(MELODY BOX) X68000	¥ 16,800

ASIC拡張関数パッケージ(B6-6301)¥9,800　C言語ライブラリ(B6-6305)¥6,800
ィスクキャッシャー(B6-6304)　¥6,800　Toys & Tools (B6-6307)¥6,800
ASIC拡張関数パッケージC言語ライブラリ付(B6-6306)　¥14,800
イコンエディタ(B6-6303)　¥4,800　CP/M68Kエミュレータ(B6-6302)¥19,800

ビデオボードを外付けに!!
ビデオボードケース(KGB-BVBX)
大好評発売中 定価9,800円

SHARPより発売されているCZ-6BV1を外付けにするケースです。このケースの使用によりあなたのX68000のスロットが開放されます。

Human68k下のソフトのCRT出力を強制的に15kHZ出力にする(768×512モード除く)おまけユーティリティ付き

全国どこでも発送可　長期クレジットOK　送料全国均一¥1,000　宅配便にて即日配送

株式会社計測技研
本社営業部/マイコンショップ/通販部　宇都宮市竹林町503—1　TEL0286-22-9811　FAX0286-25-3970
大田原営業所/マイコンショップ　大田原市美原1—13—4　TEL0287-23-5352　FAX0286-23-5364
マイコンショップ BASIC HOUSE　お申し込み・お問い合せは ☎0286-22-9811(代)

AIBIT
アイビット電子株式会社

SHARP
パソコン本体から周辺機器まで品数取り揃え
大特価セール実施中!!

型名	品名	正価	特価
PC-E500BL	ポケコン	28,800	19,500
PC-1600K	ポケコン	69,800	49,800
PC-1360K	ポケコン	36,800	32,800
PC-1360	ポケコン	29,800	19,800
PC-1262	ポケコン	24,800	19,600
PC-1248DB	ポケコン	11,000	9,800
PC-1280	ポケコン	24,800	19,600
CE-T800	ポケコンRS-232Cコンバーター	12,800	11,800
CE-203M	ポケコンRAM32K	32,000	7,000
CE-202M	ポケコンRAM16K	35,000	6,000
CE-201M	ポケコンRAM 8K	18,000	3,000
CE-1600M	ポケコンRAM32K	32,000	16,000
CE-1600F	ポケコンフロッピードライブ	39,800	34,800
CE-1600P	ポケコンプリンター	69,800	59,800
CE-1650F	ポケコンDISK	9,800	8,800
CE-161	ポケコンRAM16K	50,000	3,800
CE-1601M	ポケコンRAM64K	45,000	30,000
CE-1600E	ポケコンディスクインターフェイス	19,800	17,800
CE-158	ポケコンレベルコンバター	39,800	31,300
CE-159	ポケコンRAM 8K	35,000	4,200
CE-140T	ポケコンRS-232Cコンバター	9,800	8,800
CE-140F	ポケコンフロッピディスク	49,800	44,800
CE-123P	ポケコンプリンター	19,800	17,800
CE-120P	ポケコンプリンター	24,800	21,800
CE-126P	ポケコンプリンター	17,800	13,800
CE-124	ポケコンカセットインター	4,500	3,600
Z-VISIONplus	Z80シュミレータ デバッカー	59,800	51,000
UX-1	ホームコピーファクス	78,000	69,800
PA-9500	ハイパー電子手帳	48,000	特価
CZ-300F	X13″マイクロフロッピー	79,800	9,000
CZ-31FS	300F増設フロッピー	59,800	7,000
CZ-82F	CZ-802C増設フロッピー	59,800	6,000
CZ-501H	X1増設用ハードディスクユニット	258,000	60,000
CZ-6BS1	SCSIボード	29,800	23,800
CZ-520F	2HD/2DDミニフロッピードライブ	118,000	70,000
CZ-6BP1	数値演算ボード	79,800	63,800
CZ-6BU1	ユニバーサルI/Oボード	39,800	33,800
CZ-6BM1	MIDIボード	29,800	23,800
CZ-6BEIB	1M増設RAMボード	28,000	19,500
CZ-6BE1	1M増設RAMボード	35,000	29,500
CZ-6BE2	2M増設RAMボード	79,800	63,800
CZ-6BE4	4M増設RAMボード	138,000	110,400
CZ-6BN1	スキャナーボード	29,800	25,300
CZ-6BF1	RS-232C増設ボード	49,800	42,300
CZ-6SD1	システムラック	44,800	38,000
CZ-6TU	RRGBシステムチューナー	33,100	26,500
CZ-6BG1	X6800GPIBボード	59,800	50,000
CZ-6BC1	X6800FAXボード	79,800	67,800
CZ-6PV1	ビデオプリンター	198,000	158,000
CZ-6BV1	ビデオボード	21,000	16,800
CZ-822C	X1G MODEL30	118,000	39,800
CZ-820C	X1G MODEL10	69,800	16,800
CZ-8BGR2	グラフィックボードX1	14,800	3,000
CZ-8BF1	FDインターフェイス	14,800	11,500
CZ-8BK2	X1 漢字ROM	19,800	16,800
CZ-8BM2	232Cマウスボード	19,800	16,800
CZ-8BE2	320K外部メモリー	29,800	25,300
CZ-8BR1	立体映像セット	29,800	25,300
CZ-8BV2	カラーイメージボート	39,800	32,000
CZ-8BO1	FDインターフェイス	14,800	8,000
CZ-8TM1	X1ソフト付モデムユニット	29,800	5,000

型名	品名	正価	特価
CZ-8TM2	X1ソフト付モデムユニット	49,800	39,800
CZ-8EB3	拡張i/o box	33,800	28,000
CZ-8LM1	232cケーブル	7,200	6,000
CZ-8LM2	232cクロスケーブル	7,200	6,000
CZ-8NJ1	ジョイカード	1,700	1,360
CZ-8NT1	トラックボール	13,800	11,500
CZ-8PK10	24ドット136桁漢字プリンター	99,800	大特価
CZ-8PK7	24ドット80桁漢字プリンター	122,000	59,800
CZ-8PC5BK	48ドット熱転写カラー漢字プリンター	96,800	新発売
CZ-8BS1	X1FM音源ボード	23,800	19,800
CZ-8BK4	X1第2水準ROM	——	5,700
CZ-8NJ2	インテリジェントコントローラー	23,800	18,500
CZ-8NS1	カラーイメージスキャナー	188,000	149,000
AN-S100	アンプ付スピーカー	36,600	29,500
AN-X68	キーボードシリコンカバー	3,500	2,800
AN-X68PRO	キーボードシリコンカバー	3,500	2,800
AN-1508	ディスプレイ15P→8P変換ケーブル	——	1,600
AN-1506	ディスプレイ15P→6P変換ケーブル	——	1,600
HXD040	アイティム40Mハードディスク(ITM)	118,000	89,000
HXD140	40Mハードディスク内蔵用(ITM)	98,000	79,800
CU-14FD	カラーディスプレーアナログ0.31	74,800	49,800
MZ-1D10	12″モノクロディスプレー	41,800	25,000
MZ-1D17	15″CRT mz-5500/6500/2	124,000	59,800
MZ-1E05	MZ-2000 FDインターフェイス	24,500	18,000
MZ-1E08	プリンターI/F 2000/2200/80B	9,000	8,000
MZ-1E11	MZ-6500用 SFD I/F	38,000	25,000
MZ-1E04	MZ-2000 プリンターI/F	10,000	6,000
MZ-1E21	MZ-5500 GP I/F	36,000	12,000
MZ-1E18	MZ2000QD用インターフェイス	9,800	3,000
MZ-1E33	MZ6500パラレルI/F	34,800	28,000
MZ-1E45	MZ6500 232C I/F	50,000	15,000
MZ-1E32	MZ2500 パラレルI/F	30,000	27,000
MZ-1E44	MZ-6500 S-RN I/F	50,000	15,000
MZ-1E22	MZ-5500 GPIB I/F	72,800	25,000
MZ-1E29	RS-232Cインターフェイス300BT	17,800	9,800
MZ-1E01	MZ-3500 232Cボード	28,000	13,000
MZ-1E14	MZ1500 QD用インターフェイス	9,800	3,000
MZ-1M01	MZ-2000/2200 16ビットボード	78,000	8,000
MZ-1M09	MZ-6500 8082-2演算プロセッサ	82,000	30,000
MZ-1M03	MZ-5500 数値演算	69,000	38,500
MZ-1M12	MZ-2661 8087 演算プロセッサ	90,000	45,000
MZ-80P4B	136桁ドットプリンター	——	48,000
MZ-1P06	ドットプリンター	234,000	45,000
MZ-1P27	水平漢字プリンター	268,000	188,000
MZ-1P28	ドットプリンター漢字80桁	148,000	118,400
MZ-1P10A	24ドットプリンター漢字80桁	245,000	79,000
MZ-1P22	熱転写漢字プリンター	59,800	25,000
MZ-1P29	漢字プリンター136桁	168,000	134,400
MZ-1P30	136桁プリンター	228,000	120,000
MZ-1R01	MZ-2000/2200Gボード	39,800	10,000
MZ-1R10	MZ-5500 漢字ROM付	30,000	9,800
MZ-1R09	MZ-5500 V.RAM	35,000	15,000
MZ-1R06	MZ-5500 増設RAM	45,000	8,000
MZ-1R12	MZ-80B/2000/1500/700 RAM	35,000	8,000
MZ-1R11	MZ-5500 256KRAM	80,000	35,000
MZ-1R36	MZ-28611M増設RAM	45,000	15,000
MZ-1R35	MZ-28611M増設RAM	55,000	19,000
MZ-1R14	MZ-5500 辞書ROM	40,000	22,000
MZ-1R16	MZ-5500 128KRAM	30,000	8,000
MZ-1R27A	MZ-2500VRAM	13,000	10,000
MZ-1R21	漢字ROM	38,000	13,000

型名	品名	正価	特価
MZ-1R24	MZ-1500 辞書ROM	22,000	6,000
MZ-1R32	MZ-6500RAM	80,000	40,000
MZ-1R31	漢字ROM	28,000	20,000
MZ-1R28A	MZ-2500 辞書ROM	13,000	10,000
MZ-1R29	MZ-1P22 増設RAM17-22	32,000	12,000
MZ-1S13	MZ-1D17チルトスタンド	12,000	5,000
MZ-1T02	MZ-2200 データーレコーダー	19,800	8,500
MZ-1T03	MZ-5500 データーレコーダー	12,000	8,500
MZ-1U09	MZ-2500 拡張ボード		品切れ
MZ-1V01	パソコン FAX	278,000	85,000
MZ-1X22	モデムユニット	21,800	13,000
MZ-2Z016	MZ-5500 附属	——	5,000
MZ-2Z023	MZ-5500 GWBASIC	50,000	30,000
MZ-2Z031	MZ-6500 日本語ワープロ	49,800	15,000
MZ-2Z029	MZ-6500 TODAY	68,000	20,000
MZ-2Z064	MZ-6500 書院RAM付	69,800	28,000
MZ-2Z065	MZ-6500 書院RAMなし	49,800	15,000
MZ-2Z012	MZ-5500 附属	——	5,000
MZ-2Z013	MZ-5500 MSDOS	25,000	20,000
MZ-4Z001	MZ-5500 IBM変換	30,000	8,000
MZ-5521	本体	388,000	55,000
MZ-5511	本体	288,000	35,000
MZ-5Z013	MZ-1500 QD通信ソフト	——	3,500
MZ-6F03	ブランク QD DISK	450	400
MZ-6P18	MZ-1P18、28カットシートフィーダー	60,000	35,000
MZ-6P29	MZ-1P29 カットシートフィーダー	50,000	37,500
MZ-6P27	MZ-1P27 カットシートフィーダー	58,000	39,800
MZ-6P06	MZ-1P06 トラクターフィード	15,000	7,500
MZ-6P20	MZ-1P22/17ロールホルダー	3,000	2,700
MZ-6Z22	MZ-6500(50)CP/M86BASIC-3	10,000	6,000
MZ-6Z25	M-50 ストリーマユーティリティZプロセッサ	39,800	15,000
MZ-80T20A	MZ-80 マシンランゲージ	6,000	5,000
MZ-80TUB	MZ-80 バックアップ	20,000	8,000
MZ-80TU	MZ-80 システムプログラム	20,000	8,000
MZ-80T40A	MZ-80 PASCAL	10,000	5,000
MZ-80T70A	MZ-80 FDOS	20,000	7,000
MZ-8BGK	MZ-80 BGRAM2	39,000	10,000
MZ-8B104	MZ200/2200 GP IBインターフェイス	45,000	18,000
MZ-8BC01	MZ200/2200 GP IBケーブル	18,000	8,000
UE-1U01	X286L スロットBOX	5,000	4,000
UE-1R02	4M RAMボード	300,000	240,000
UE-1R06	辞書ROMボート	32,800	25,600
UE-1R01	2M RAMボード	160,000	128,000
UE-1R05	拡張グラフィックボード	92,000	55,000
UE-1R03	1M RAMボード	100,000	80,000
UE-1R04	2M RAMボード	180,000	144,000
UE-1P03	80桁漢字プリンタ	——	特価
UE-1P04	136桁漢字プリンタ	——	特価
UE-1P05	136桁漢字水平プリンタ	——	特価
UE-1P02	高速136桁漢字プリンタ	550,000	440,000
UE-1P01	136桁漢字プリンタ	268,000	214,400
UE-1E04	S-RNインターフェイスカード	70,000	56,000
UE-1E02	AX286L ICカードI	45,000	36,000
UE-1E03	5″FDインターフェイスカード	28,000	22,400
UE-1D03	15インチカラーディスプレイ	123,000	98,400
UE-1D02	14インチカラーディスプレイ	158,000	126,400
IO-735X	カラープリンター	248,000	195,000
BF-68PRO	フィルター	19,800	16,800
X6800キーボード延長ケーブル(1.5m)		2,500	2,000
ディスプレーケーブルアナログ15P(3m)		5,000	4,000
ディスプレーケーブルアナログ15P(1.5m)		4,300	3,500

ポケコン関係周辺機器サプライ製品及シャープ関係のソフトウエア全種取扱います。

FM TOWNS/FM NOTE/東芝ダイナブック、周辺機器も取扱っております。

満開の電子(でんこ)ちゃん

作：いわいいっぺい
え：岡村 祭

佐藤 一秀（愛知県）

グラフィックオンリーだった私は、当初「電脳倶楽部なんてCGしか見るものがないなー」と思っていたのですが、X68Kをさわっているうちに中のツールの使い方が判り、だんだん電クラが面白くなっていきました（やはりイラストばかりかいていては駄目だな）。そしてMFGEDやCRAYON・X等のグラフィックツール、その他グラフィック派にとって美味しいツールが何と沢山あることか！今では、電クラを購読していてホントに良かったと思っています(これで一ヶ月千円は安い、買いなさい！)。

ACCESS

X1エミュレータ

好評発売中

定価¥9,800

X1エミュレータはX68000上でX1シリーズのアプリケーションを実行するためのソフトエミュレータです。X1のアプリケーションを完全にソフトウェアのみでエミュレートしているため、X1上での実行速度と比較して、平均3〜5倍程度おそくなりますが、X68000のマシン上に実現した仮想X1マシンを楽しめます。また、X1とX68000の相互間でファイルを転送するためのユーティリティと専用ケーブルが付属しますので、X1上で作り上げたソフトの資産をX68000上に移行することも簡単にできます。

X1エミュレータの機能

- X1エミュレータはX1に相当する機能をエミュレート。この仮想コンピュータには最大4つのドライブが仮想的に接続。
- X1エミュレータからみたドライブはHuman68kのドライブ上にあるファイルで仮想的に実現。このファイルはX1用の5" 2Dディスクのイメージをファイル転送ユーティリティでまるごと転送したもの。
- X1エミュレータで仮想的に実現したX1は仮想ドライブから起動。このため仮想ドライブ用ファイルには、X1を立ち上げるために必要なHuBASICやCP/Mなどのシステムプログラムが必要。
- X1エミュレータでは、X1の持つVRAMを含むメモリイメージやZ80CPUを仮想的にソフトウェアで実現。

ファイル転送ユーティリティ

ディスク転送

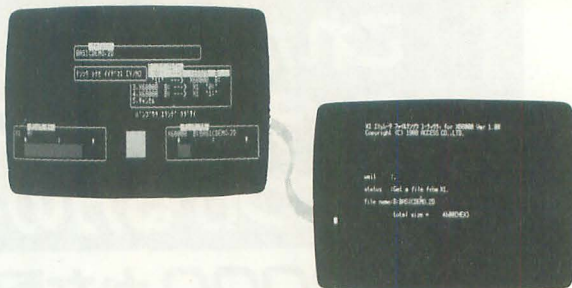
X1ディスク⟷X68000 Human68k(5" 2Dディスクイメージファイル)

- X1エミュレータではHuman68k上のディスクイメージファイルを仮想ドライブとして使用。

ファイル転送

X1 BASIC:CP/M⟷X68000 Human68k

- X1で作ったプログラム&データをX68000上で使用。

※付属の専用ケーブルをX1とX68000に接続してファイルを転送します。

X1エミュレータ Q&A

Q. ファイル転送のために別途RS-232Cケーブルを買わないといけないのですか?

A. 専用のケーブルが付属しますのでその必要はありません。

Q. X1BASICのプログラムをX68000上のX-BASICで使えますか?

A. 通常のセーブではコードが違うので使用できませんが、アスキーセーブしたファイルであればX-BASIC上でそのままロード可能です。

Q. TurboBASICで作成した住所録などの漢字を含んだデータがあるのですがX68000上にファイル転送できますか?

A. X1TurboもX68000も漢字はシフトJISコードなのでファイルの転送は可能です。ただし、漢字ROMを必要とするものはサポートしていません。

Q. Turbo用のソフトは動きますか?

A. X1用のみでTurbo専用のソフトは動きません。

Q. ゲームは動きますか?

A. 純粋にBASICでかかれたものは動きますが、プロテクトがかかったものや直接ハードをアクセスするような市販のゲームは動きません。

*タイミング等ハードウェアに依存するようなソフトは、原理上実行できない、もしくは正常に動作しない場合がありますのでご注意ください。
*一部サポートしていない機能があります。

X1エミュレータ通信販売 購入希望として住所、氏名、電話番号をお知らせください。注文書をお送り致します。

*この商品価格には消費税は含まれておりません。
*CP/Mはデジタルリサーチ社の商標です。
文中のソフトウェアは各社の商標です。
*製品の仕様、名称は予告なく変更する場合もございますのであらかじめご了承ください。

有限会社アクセス 〒101 東京都千代田区神田神保町1-64 神保町協和ビル7F
☎03(3233)0200(代) FAX.03(3291)7019

資料請求券 oh!X 4月号

な人、
ぜひおいで〜
が大好きな人・
心者・マニア、プログラマーから
さん・ただのミーハーに
好き・大酒のみから
何かをやるのが大好きな人まで。
それと、作品の発表の場を
求めている人！

パソコン/ワープロ通信ネットワークサービス

J&P HOT LINE

SIG探訪PART 9

CZ−CLUB

(ジャンプコード：CZCLUB)

PAXT.T(香川県)

案内豚王那乃(神奈川県)

なんばに集合して、日本橋へと向かいつつ中古ショップを覗き、
我らが本拠J&Pテクノランドにて一服、正面玄関で記念写真を撮り、
宴会OPを先頭にて大宴会！自分のマシンに関する情報が飛びかい、
ワイワイ騒ぎ酔いが足らんとR360で大回転、そんなこんなで新生
CZ−CLUB初のOFFは終わりました。これからも年2回は
OFFをやりたいと思います。初めての方もどんどん参加して下さいね。

1日30分の参加で屈指の最強マシンに?!

「シャープのXファミリー」ユーザーが、その情報交流のために集まっているＳＩＧなので、その筋の情報にはとにかく強い。プログラマー・ハード屋……といったプロフェッショナルはもちろん、マニアから初心者まで幅広いメンバーがお互いの情報を大切にやり取りしています。
68KのMDXやＸ１のMUSなど音楽関係やその他の便利なツール、ミュージックデータや面白いサンプリング音、豊富なPDSがいっぱいです。初心者の方にはQ&Aコーナーもあるので心強いことこの上なし。
巷のうわさでは、１日に30分CZ−CLUBに参加するだけで、どんな(?)マシンも日本屈指の最強マシンに変身するとか……。

その他 楽しいメニューがまだまだいっぱい！

★Ｊ＆Ｐならではのパソコン・家電製品の会員割引もある**ONLINE SHOPPING**。
★Ｊ＆Ｐだから強い!!パソコン情報をはじめとする役に立つ**DATA BASE**。
★みんなでおしゃべり**オンライントーク**(CHAT機能)。
★地域別・テーマ別ボードで充実の**BBS**(電子掲示板)。
★ビジュアルデータもばっちり送受信できる**X-MODEM**。

J&P HOT LINEへのご入会はスタータキットで。

お求めは、下記のお店へ。又は現金書留にて、¥3,000＋¥90(消費税3%)＝¥3,090を事務局までお送り下さい。
すぐにスタータキットをお送りします。

お問い合わせは 〒556 大阪市浪速区日本橋西1-6-5 上新電機株式会社
J&P HOT LINE事務局宛　TEL.(06)632-2521

スタータキットのお求めはJ&P各店でどうぞ。

店名	住所	電話
谷　店	東京都渋谷区道玄坂2丁目28番4号	☎(03)3496-4141
田　店	東京都町田市森野1丁目39番16号	☎(0427)23-1313
王子店	東京都八王子市旭町1番1号八王子そごう7F	☎(0426)26-4141
川　店	東京都立川市幸町4−39−1	☎(0425)36-4141
厚木店	厚木市中町3−4−3	☎(0462)25-1548
山　店	富山市桜町2−1−10	☎(0764)32-3133
沢　店	金沢市入江2−63	☎(0762)91-1130
地　店	金沢市寺地2−3	☎(0762)47-2524
大須店	名古屋市中区大須4丁目2-48	☎(052)262-1141
テクノランド	大阪市浪速区日本橋5丁目6番7号	☎(06)634-1211
メディアランド	大阪市浪速区日本橋5丁目8番26号	☎(06)634-1511
コスモランド	大阪市浪速区難波中2丁目1番17号	☎(06)634-3111
U. S. LAND	大阪市浪速区日本橋4丁目9番15号	☎(06)634-1411
ビジネスランド	大阪市北区梅田1-1-3大阪駅前第3ビルB2	☎(06)348-1881
梅田店	大阪市北区小松原町1−10	☎(06)362-1141
高槻店	高槻市高槻町11番16号	☎(0726)85-1212
くずは店	枚方市楠葉花園町15番2号	☎(0720)56-8181
千里中央店	豊中市新千里東町1-3 SENCHU PAL 2番街4F	☎(06)834-4141
摂津富田店	高槻市大畑町24−10	☎(0726)93-7521
寝屋川店	寝屋川市緑町4−20	☎(0720)34-1166
藤井寺店	藤井寺市岡2丁目1番33号	☎(0729)38-2111
岸和田店	岸和田市土生町2451−3	☎(0724)37-1021
さんのみや1ばん館	神戸市中央区八幡通3-2-16	☎(078)231-2111
西宮店	兵庫県西宮市河原町5−11	☎(0798)71-1171
姫路店	姫路市東延末1丁目1番住友生命姫路南ビル1F	☎(0792)22-1221
京都寺町店	京都市下京区寺町通仏光寺下ル恵比須之町549	☎(075)341-3571
京都近鉄店	京都市下京区烏丸通七条下ル東塩小路町702	☎(075)341-5769
和歌山店	和歌山市元寺町4丁目4番地	☎(0734)28-1441
奈良1ばん館	奈良市三条町478−1	☎(0742)27-1111
郡山インター店	大和郡山市横田693−1	☎(07435)9-2221
熊本店	熊本市手取本町4−12	☎(096)359-7800

T4910217904564 雑誌 02179-4